UNIVERSIDADE POSITIVO
André Luís Strechar do Nascimento
Jhon Alisson Gonçalves dos Santos

DESENVOLVIMENTO DE SOFTWARE PARA PRÉ-DIMENSIONAMENTO E DETERMINAÇÃO DA SOLUÇÃO MAIS ECONÔMICA DE FUNDAÇÕES EM ESTACAS UTILIZANDO O MÉTODO DE DÉCOURT-QUARESMA

CURITIBA
2016

André Luís Strechar do Nascimento

Jhon Alisson Gonçalves dos Santos

DESENVOLVIMENTO DE SOFTWARE PARA PRÉ-DIMENSIONAMENTO E DETERMINAÇÃO DA SOLUÇÃO MAIS ECONÔMICA DE FUNDAÇÕES EM ESTACAS UTILIZANDO O MÉTODO DE DÉCOURT-QUARESMA

Trabalho de Conclusão apresentado ao curso de Engenharia Civil da Universidade Positivo como parte dos requisitos para graduação.

Orientador: Prof. Juliano Jorge Scremin

CURITIBA

2016

RESUMO

Neste trabalho são apresentados conceitos fundamentais e procedimentos indispensáveis para a criação de um software que pré-dimensiona fundações profundas em estacas. O software desenvolvido considera a utilização de diferentes perfis de sondagens geotécnicas, definidos pelo usuário, e fornece como resultados as dimensões de seções transversais, comprimentos embutidos, a quantidade de estacas, por bloco de coroamento e o custo estimado para diferentes tipos de estacas selecionadas para o estudo em questão. O programa foi desenvolvido utilizando a linguagem VB.NET empregando a plataforma do Visual Studio. As saídas obtidas permitem um comparativo do volume de material gasto para cada tipo de solução em estacas cogitado (por exemplo: cravada, escavada, hélice contínua e etc.) visando determinar qual solução apresenta-se mais econômica. Para realizar o dimensionamento da capacidade de carga foi utilizado o método semi-empírico Décourt-Quaresma, cuja escolha se deu devido a este embasar-se nos resultados de ensaios de sondagem à percussão (SPT) que é o ensaio mais comumente utilizado no Brasil.

Palavra Chave: Décourt – Quaresma, Capacidade de carga, Código computacional.

SUMÁRIO

1 Introdução 9

1.1 Justificativa 10

1.2 Objetivo Geral 11

1.3 Objetivo Específico 11

2 Fundamentação teórica 12

2.1 Considerações gerais sobre fundações profundas 12

2.2 Sondagem a percussão - SPT (Standard Penetration Test) 12

2.3 Capacidade de carga em fundações profundas 14

2.4 Capacidade Estrutural 16

2.5 Método de Décourt-Quaresma 17

2.6 Aspectos gerais sobre as estacas empregadas no método semi-empírico de Décourt-Quaresma 20

2.6.1 Estaca Pré-fabricada de concreto 20

2.6.2 Estaca escavada com lama 21

2.6.3 Estaca Hélice contínua 22

2.6.4 Estaca raiz 24

2.6.5 Microestaca (injetada sob alta pressão) 26

2.6.6 Estaca escavada sem lama (Broca) 27

2.7 Estaqueamento 28

2.8 Visual Basic 30

2.9 Banco de dados – XML 31

2.10 Revisão Bibliográfica 31

3 Procedimentos Metodológicos 33

3.1 Fluxograma do *CALFES* 33

3.2 Entrada de dados _____34
3.3 Dados armazenados _____35
3.4 Função: CALCULA_AREA_PERIMETRO_ESTACAS _____35
3.5 Função: CALCULAR _____36
3.6 Função: DECOURT_QUARESMA _____37
3.7 Função: SOLUCAO_1P _____39
3.8 Função: SOLUCAO_TODOS _____42
3.9 Saída de dados _____43
3.10 Validação do Software _____44
4 Resultados _____45
5 Ánalise dos Resultados _____51
6 Considerações Finais _____53
Referências _____54
Apêndice A (Código Computacional) _____56
Apêndice B (Validação do Software) _____101

LISTA DE TABELAS

TABELA 1 - PARÂMETROS PARA DIMENSIONAMENTO. 16

TABELA 2 – VALORES PARA O COEFICIENTE k. 17

TABELA 3: COEFICIENTES α E β PARA AS ESTACAS. 18

TABELA 4 – RESULTADOS A PARTIR DE CÁLCULOS MANUAIS. 50

TABELA 5 – FURO DE SONDAGEM. 101

TABELA 6 – DEFINIÇÕES DAS PROPRIEDADES DAS ESTACAS – HÉLICE CONTÍNUA. 102

TABELA 7 – DEFINIÇÕES DAS PROPRIEDADES DOS BLOCOS DE 1 ESTACA. 102

TABELA 8 – DEFINIÇÕES DAS PROPRIEDADES DOS BLOCOS DE 2 ESTACA. 102

TABELA 9 – DEFINIÇÕES DAS PROPRIEDADES DOS BLOCOS DE 4 ESTACA. 103

TABELA 10 – DEFINIÇÕES DAS CARGAS. 103

TABELA 11 – RESULTADOS DA CAPACIDADE DE CARGA GEOTÉCNICA PARA DIÂMETRO DE 40cm – PARTE 01. 104

TABELA 12 – RESULTADOS DA CAPACIDADE DE CARGA GEOTÉCNICA PARA DIÂMETRO DE 40cm – PARTE 02. 105

TABELA 13 – RESULTADOS DA CAPACIDADE DE CARGA GEOTÉCNICA PARA DIÂMETRO DE 50cm – PARTE 01. 106

TABELA 14 – RESULTADOS DA CAPACIDADE DE CARGA GEOTÉCNICA PARA DIÂMETRO DE 50cm – PARTE 02. 107

TABELA 15 – RESULTADOS DA CAPACIDADE DE CARGA GEOTÉCNICA PARA DIÂMETRO DE 60cm – PARTE 01. 108

TABELA 16 – RESULTADOS DA CAPACIDADE DE CARGA GEOTÉCNICA PARA DIÂMETRO DE 60cm – PARTE 02. 109

TABELA 17 – SOLICITAÇÕES MÁXIMAS E MÍNIMAS – PILAR P1. 109

TABELA 18 – SOLICITAÇÕES MÁXIMAS E MÍNIMAS – PILAR P2. 110

TABELA 19 – SOLICITAÇÕES MÁXIMAS E MÍNIMAS – PILAR P3. 110

TABELA 20 – CAPACIDADE ESTRUTURAL A COMPRESSÃO E A TRAÇÃO. 110

TABELA 21 – EMBUTIMENTOS MÍNIMOS PARA BLOCO1. 111

TABELA 22 – EMBUTIMENTOS MÍNIMOS PARA BLOCO2. 111

TABELA 23 – EMBUTIMENTOS MÍNIMOS PARA BLOCO4-1. 111

TABELA 24 – EMBUTIMENTOS MÍNIMOS PARA BLOCO4-2. 112

TABELA 25 – RESULTADOS. 112

LISTA DE FIGURAS

FIGURA 1 – ENSAIO SPT 13

FIGURA 2 – PROVA DE CARGA ESTÁTICA COM ESTRUTURA METÁLICA REAGINDO AO SISTEMA HIDRÁULICO. 15

FIGURA 3 – PROVA DE CARGA ESTÁTICA COM TIRANTES REAGINDO AO SISTEMA HIDRÁULICO. 15

FIGURA 4 – ESTACAS PRÉ-FABRICADAS DE CONCRETO. 21

FIGURA 5 – ESTACA ESCAVADA COM A PRESENÇA DE LAMA BENTONÍTICA. 22

FIGURA 6 – ESTACAS HÉLICE CONTÍNUA.....24

FIGURA 7 – ESTACA RAIZ. 25

FIGURA 8 – MICROESTACA.....27

FIGURA 9 – ESTACA BROCA COM TRADO MANUAL. 28

FIGURA 10 – ESTACA BROCA COM TRADO MECÂNICO. 28

FIGURA 11 – ESTAQUEAMENTO COM CARGAS VERTICAIS E MOMENTOS FLETORES EM AMBAS AS DIREÇÕES. 29

FIGURA 12 – FLUXOGRAMA GERAL DO FUNCIONAMENTO DO SOFTWARE. ..33

FIGURA 13 – FLUXOGRAMA DA ENTRADA DE DADOS. 34

FIGURA 14 – FLUXOGRAMA DA FUNÇÃO: CALCULA_AREA_PERIMETRO_ESTACAS. 36

FIGURA 15 – FLUXOGRAMA DA FUNÇÃO: CALCULAR. 36

FIGURA 16 – FLUXOGRAMA DE FUNCIONAMENTO DA FUNÇÃO: DÉCOURT-QUARESMA. 38

FIGURA 17 – FLUXOGRAMA DA FUNÇÃO: SOLUCAO_1P – PARTE 01. 39

FIGURA 18 – FLUXOGRAMA DA FUNÇÃO: SOLUCAO_1P – PARTE 02. 40

FIGURA 19 – FLUXOGRAMA DA FUNÇÃO: SOLUCAO_TODOS. 42

FIGURA 20 – FLUXOGRAMA DA SAÍDA DE DADOS. 43

FIGURA 21 – FURO DE SONDAGEM 01 – PARTE 01. 45

FIGURA 22 – FURO DE SONDAGEM 01 – PARTE 02. 46

FIGURA 23 – PROPRIEDADES GEOMÉTRICA DAS ESTACAS HÉLICE CONTÍNUA.....46

FIGURA 24 – ENTRADA DE DADOS PARA BLOCO COM 1 ESTACA. 47

FIGURA 25 – ENTRADA DE DADOS PARA BLOCO COM 2 ESTACA. 47

FIGURA 26 – ENTRADA DE DADOS PARA BLOCO COM 4 ESTACA. 48

FIGURA 27 – PARÂMETROS PARA O DIMENSIONAMENTO. 48

FIGURA 28 – CARGAS DOS PILARES E CALCULOS PARA ESTACA HÉLICE CONTÍNUA.....49

FIGURA 29 – RELATÓRIO COM RESULTADOS GERADOS PELO SOFTWARE. .50

1 INTRODUÇÃO

As edificações estão sujeitas a vários tipos de ações, como o peso próprio da estrutura, cargas acidentais e excepcionais. Estas estruturas sofrem esforços que deverão ser resistidos pelos elementos estruturais tais como lajes, vigas, pilares e os elementos de fundação (CARVALHO; DE FIGUEIREDO FILHO, 2004).

Os elementos de fundação são responsáveis por transmitir os esforços gerados pela edificação para o solo, ora pela base, ora pela superfície lateral ou em uma combinação entre elas (NBR 6122, 2010).

Para a definição de como será composta a fundação, foram criados métodos teóricos e semi-empíricos para a determinação da capacidade de carga para um elemento de fundação isolado (VELLOSO; LOPES, 2011).

Dentre os métodos teóricos estão os métodos de Terzaghi, a solução de Meyerhof, a solução de Vesic, entre outros, sendo que estes se baseiam em soluções teóricas de capacidade de carga e parâmetros do solo. Já entre os métodos semi-empíricos, estão os métodos de Décourt-Quaresma, Aoki e Velloso, Teixeira, Vorcaro-Velloso, entre outros, que se baseiam em correlações teóricas e em ensaios de penetração, estes ensaios são o CPT (Cone Penetration Test) e o ensaio SPT (Standard Penetration Test), sendo o último o mais difundido no Brasil (VELLOSO; LOPES, 2011).

Com o passar dos anos, para ter mais agilidade no processo de dimensionamento de estacas, foram criados Softwares e Planilhas eletrônicas, que incorporam os métodos de dimensionamento. Estas ferramentas, foram geradas para minimizar possíveis erros e o consumo de tempo necessário para a determinação da capacidade de carga, através de cálculos manuais. Desta maneira, pode-se testar diversas soluções, para a mesma obra, afim de obter a solução que contemple aspectos técnicos como a possibilidade de execução mediante o nível do lençol freático e tipo de solo do local, como também, aspectos econômicos relativos a disponibilidade e custo de materiais e equipamentos para execução da solução (VELLOSO; LOPES, 2011).

Assim sendo, a determinação do tipo de estaca a ser utilizada em uma fundação profunda, em muitas vezes, requer uma comparação entre duas ou mais alternativas de soluções possíveis para a mesma obra em que há a disponibilidade do material e dos equipamentos correlatos. De modo a poder orçar as alternativas tecnicamente possíveis, faz-se necessário um levantamento, mesmo que aproximado, dos volumes de material e comprimentos de cravação / escavação para cada uma das soluções cogitadas.

Para viabilizar esta quantificação dos insumos faz-se necessário o cálculo das quantidades de estacas por bloco com seus respectivos comprimentos embutidos e dimensões de seção transversal para cada alternativa aventada.

Este trabalho propõe-se a desenvolver, com base na NBR 6122:2010 e na bibliografia apresentada a seguir, um software capaz de determinar a capacidade de carga de uma estaca isolada, o menor consumo de material, todas as soluções possíveis para o mesmo pilar, a quantidade de estacas necessária para suportar os esforços solicitantes e comparar o custo de insumos de todas as soluções.

1.1 Justificativa

Para o cálculo da capacidade geotécnica resistente de estacas são muito utilizados o auxílio de planilhas eletrônicas, que auxiliam nesta tarefa. Apesar de auxiliarem no processo de cálculo a comparação da solicitação axial na estaca com a capacidade geotécnica resistente acaba sendo manual, tornando o processo muito lento.

Esses softwares não comparam a solicitação solicitante com a geotécnica, nem buscam o menor custo possível para diferentes tipos de estacas.

Dentre os trabalhos com temas correlatos, pode-se destacar a dissertação de mestrado com o tema, “Otimização dos custos dos projetos de estaqueamento através de uma planilha de cálculo eletrônica de fundações profundas” (Carvalho, 2014), que calcula os esforços atuantes nas estacas e considera blocos de fundação previamente estabelecidos. Porém, a planilha não compara resultados, ela apenas calcula para a solução, referente aos dados inseridos pelo usuário, todavia esta solução pode não ser a mais econômica.

Outro trabalho que merece destaque, é o artigo publicado na Revista Eletrônica de Engenharia Civil, com o tema “Rotina computacional para a previsão da capacidade de carga em estacas”, esta planilha apresenta um gráfico que representa a capacidade de carga ao longo do fuste e faz a comparação dos resultados obtidos entre 3 métodos propostos, a saber: Décourt-Quaresma, Aoki-Velloso e Teixeira. Entretanto, esta planilha não apresenta a solução mais econômica, pois calcula apenas para a solução fornecida pelo usuário, podendo não ser a melhor opção entre muitas (FERREIRA; DELALIBERA; SILVA, 2014).

Sendo assim, tendo em vista a agilização do processo de cálculo e a procura pela solução mais econômica, propõe-se a criação de um software capaz de calcular todos os tipos de estaca, incorporados no método de Décourt-Quaresma, para um mesmo conjunto de cargas, podendo dar ao usuário a opção de escolher a estaca que melhor se adequa a situação estudada. Além destas análises, o usuário poderá inserir parâmetros que sejam padrões, como: o embutimento, onde o usuário poderá escolher um comprimento padrão para os diferentes tipos de estacas, o rótulo, em que o usuário define uma única seção transversal para as diferentes cargas, e a quantidade de estacas, na qual o usuário define uma quantidade padrão de estacas para as diferentes análises. Por fim o usuário poderá gerar um relatório com o resumo das estacas, para poder quantificar em volume de material.

1.2 Objetivo Geral

Desenvolver um software para pré-dimensionamento de fundações profundas em estacas, utilizando o método semi-empírico de Décourt-Quaresma.

1.3 Objetivo Específico

Os objetivos específicos pertinentes ao trabalho são:

- Indicar a solução em estacas que apresenta o menor custo;
- Apresentar relatórios gerados pelo software com o custo total de todas as soluções estudadas pelo usuário, com todos os dados fornecidos e os resultados calculados.

2 FUNDAMENTAÇÃO TEÓRICA

Para este trabalho, foram utilizadas algumas bibliografias que ajudarão com a fundamentação e o desenvolvimento do Software. A seguir serão apresentados conceitos sobre fundações profundas e métodos computacionais.

2.1 Considerações gerais sobre fundações profundas

Segundo a NBR 6122:2010, as fundações profundas são os elementos que transmitem a carga ao terreno, pela base (resistência de ponta), por sua superfície lateral (resistência lateral) ou por uma combinação entre as duas ações, devendo: estar com sua ponta ou base em uma profundidade superior ao dobro de sua menor dimensão em planta e no mínimo a 3,0m de profundidade. Neste tipo de fundação estão inclusas as estacas, tubulões e caixões. As estacas se diferenciam dos tubulões e caixões, pois sua execução é apenas por equipamentos e ferramentas, não necessitando de operários em seu interior, em nenhuma fase.

A escolha de uma fundação deve ser feita após verificar as condições técnicas e econômicas. Para tanto devem ser conhecidos os seguintes elementos:

- Proximidade dos edifícios limítrofes, bem como seu tipo de fundação e estado da mesma;
- Natureza e características do subsolo no local da obra;
- Grandeza das cargas a serem transmitidas à fundação;
- Limitação dos tipos de fundação existentes no mercado.

O problema pode ser resolvido por eliminação, escolhendo, entre os tipos de fundação existentes, aqueles que satisfaçam tecnicamente cada caso em questão (ALONSO, 1983).

2.2 Sondagem a percussão - SPT (Standard Penetration Test)

O Standard Penetration Test é reconhecidamente a mais popular e econômica ferramenta de investigação do subsolo. Métodos rotineiros de projeto de

fundações diretas e profundas usam sistematicamente os resultados do ensaio SPT, especialmente no Brasil (SCHNAID, 2000).

No Brasil as sondagens a percussão de simples reconhecimento com SPT, são regidas pela NBR 6484:2001, que tem por objetivo prescrever o método de execução das sondagens de simples reconhecimento destas.

As vantagens deste ensaio são (SCHNAID, 2000):

- Simplicidade do equipamento;
- Baixo custo;
- A obtenção de um valor numérico que pode ser relacionado com métodos empíricos de projeto, este valor denomina-se índice de resistência a penetração, que indicará a qualidade das camadas amostradas.

Abaixo a figura 1 ilustra esquematicamente o ensaio SPT:

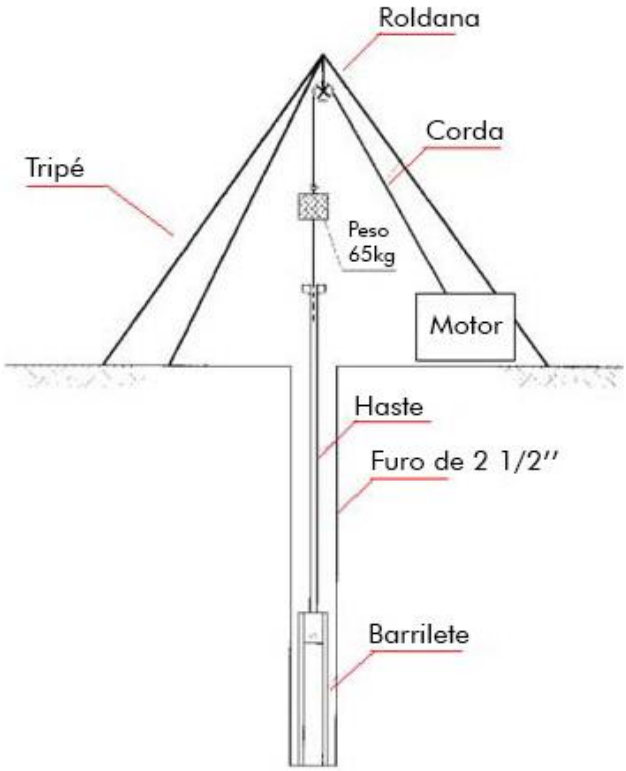


FIGURA 1 – ENSAIO SPT
FONTE: Schnaid (2000).

O ensaio é realizado com um amostrador cravado por um martelo de 65 kg em queda livre, a uma altura de 75 cm. Durante este procedimento são registrados o

número de golpes necessários à penetração de cada 15 cm da camada explorada, além das propriedades e características do solo coletado pelo amostrador (SCHNAID, 2000).

Após o término do ensaio, é gerado o boletim de sondagem, contendo: as descrições das camadas de solo, o índice de resistência à penetração, gráfico de resistência x profundidade, o limite da sondagem por furo e a presença ou não de lençol freático (SCHNAID, 2000).

Os índices de resistência à penetração servem de indicação qualitativa à previsão de problemas, por exemplo, solos com N_{spt} maiores que 30 indicam um tipo de solo resistente e estável, sem necessidade de estudos geotécnicos mais elaborados. Já os solos com N_{spt} menores que 5 indicam solos pouco resistentes, ocasionando estudos mais refinados (SCHNAID, 2000).

2.3 Capacidade de carga em fundações profundas

Uma estaca submetida a uma carga vertical irá resistir a esta solicitação, uma parcela pela resistência lateral e outra pela resistência de ponta (HACHICH, 1998).

A capacidade de carga Q_u é definida como a soma das cargas máximas que podem ser suportadas pelo atrito lateral e pela ponta, Q_s e Q_p, respectivamente:

$$Q_u = Q_s + Q_p \tag{1}$$

Nomeando por q_s e q_p, as tensões limites de cisalhamento ao longo do fuste e normal ao nível da base e considerando A_s e A_p, como sendo, respectivamente, a área lateral do fuste e a área da seção transversal da estaca, tem-se a seguinte expressão (HACHICH, 1998):

$$Q_u = Q_s + Q_p = q_s \times A_s + q_p \times A_p \tag{2}$$

A capacidade de carga pode ser definida através do ensaio de prova de carga estática. Este ensaio consiste em reproduzir o comportamento da solicitação de uma estaca. Para a realização deste ensaio, utiliza-se uma placa metálica, que é carregada por um macaco hidráulico que reage contra uma caixa carregada, uma

estrutura metálica ou qualquer outro material que possa servir como uma carga de reação ou pode reagir contra um sistema de tirantes (ALONSO, 1983).

A figura 2 ilustra este ensaio com uma estrutura metálica reagindo ao macaco hidráulico.



FIGURA 2 – PROVA DE CARGA ESTÁTICA COM ESTRUTURA METÁLICA REAGINDO AO SISTEMA HIDRÁULICO.
FONTE: SPFE – Sociedade Portuguesa de Fundações Especiais LDA.

A figura 3 ilustra este ensaio com tirantes reagindo ao macaco hidráulico.



FIGURA 3 – PROVA DE CARGA ESTÁTICA COM TIRANTES REAGINDO AO SISTEMA HIDRÁULICO.
FONTE: Costa Fortuna – Engenharia de Fundações.

2.4 Capacidade Estrutural

Segundo a NBR 6122:2010, as estacas ou tubulões, quando solicitados a cargas de compressão e tensões, limitadas aos valores da Tabela 1, podem ser executadas sem a presença de armadura, a não ser quando exista armadura de ligação com o bloco.

Tipo de estaca	f_{ck} máximo de projeto (Mpa)	γ_f	γ_c	γ_s	Comprimento útil mínimo (incluindo trecho de ligação com o bloco) e % de armadura mínima		Tensão média atuante abaixo da qual não é necessário armar (exceto ligação com o bloco) (Mpa)
					Armadura (%)	Comprimento (m)	
Hélice/hélice de deslocamento	20	1,4	1,8	1,15	0,5	4,0	6,0
Escavadas sem fluído	15	1,4	1,9	1,15	0,5	2,0	5,0
Escavadas com fluído	20	1,4	1,8	1,15	0,5	4,0	6,0
Strauss	15	1,4	1,9	1,15	0,5	2,0	5,0
Franki	20	1,4	1,8	1,15	0,5	Armadura integral-	-
Tubulões não encamisados	20	1,4	1,8	1,15	0,5	3,0	5,0
Raiz	20	1,4	1,6	1,15	0,5	Armadura integral	-
Microestaca	20	1,4	1,8	1,15	0,5	Armadura integral	-
Estaca trado vazado segmentado	20	1,4	1,8	1,15	0,5	Armadura integral	-

TABELA 1 - PARÂMETROS PARA DIMENSIONAMENTO.
FONTE: NBR 6122 (2010).

O dimensionamento estrutural deve ser feito de acordo com a NBR 6118:2014. A carga estrutural admissível pode ser calculada pela seguinte equação:

$$Q_k = \frac{1}{\gamma_f} \times \left(\frac{0{,}85 \times f_{ck} \times A_c}{\gamma_c} + \frac{f_{yk} \times A_s}{\gamma_s} \right) \quad (3)$$

Onde:

- Q_k: Carga estrutural admissível;
- f_{ck}: Resistência característica do concreto a compressão;
- A_c: Área da seção transversal;
- f_{yk}: Resistência característica do aço a tração;
- A_s: Taxa de armadura (Tabela 1);
- γ_f: Fator de majoração das cargas (Tabela 1);
- γ_c: Fator de minoração da resistência do concreto (Tabela 1);
- γ_s: Fator de minoração da resistência do aço (Tabela 1).

2.5 Método de Décourt-Quaresma

O método de Décourt-Quaresma consiste em calcular a resistência de ponta e o atrito lateral na estaca, utilizando coeficientes conhecidos e tabelados para cada tipo de solo e cada tipo de estaca que o método é composto (VELLOSO; LOPES, 2011).

Para cálculo da resistência de ponta unitária, utiliza-se a seguinte equação:

$$q_{p,ult} = k.N_1 \quad (4)$$

Onde:

- $q_{p,ult}$: Resistência de ponta unitária;
- k: Coeficiente que depende do tipo de solo na ponta (Tabela 2).
- N_1: Valor resistente à penetração da estaca no solo (valores da sondagem).

Para obter o valor de N_1, deve-se utilizar os valores correspondentes à ponta da estaca, o imediatamente anterior e o imediatamente posterior. Com os valores de k, utilizando a tabela 2, obtêm-se o valor da resistência de ponta em tf/m².

Tipo de solo	k (tf/m²)
Argilas	12
Siltes argilosos (alteração de rocha)	20
Siltes arenosos (alteração de rocha)	25
Areias	40

TABELA 2 – VALORES PARA O COEFICIENTE k.
FONTE: Velloso e Lopes (2011).

Na versão inicial do método Décourt-Quaresma, considerava-se os valores de N ao longo do fuste, sem levar em consideração os valores utilizados para cálculo da resistência de ponta, através da média obtida pelos valores dos índices de penetração, obtinha-se o atrito ao longo do fuste.

Em sua primeira versão o método era utilizado apenas para estacas pré-moldadas de concreto, podendo-se admitir, por meio de aproximação, as estacas Franki e Strauss (VELLOSO; LOPES, 2011).

Na segunda versão, Décourt e Quaresma aperfeiçoaram o método, no que se diz respeito ao atrito lateral unitário, em tf/m², que é dado pela equação:

$$q_{l,ult} = \left({}^{N_2}/_{3}\right) + 1 \qquad (5)$$

Onde:

- $q_{l,ult}$: Atrito lateral unitário;
- N_2: é a média dos valores de N ao longo do fuste, desconsiderando valores utilizados para cálculo do N_1.

Deve-se observar que valores menores que 3 devem ser considerados iguais a 3 e valores acima de 50 devem ser considerados 50 (VELLOSO; LOPES, 2011).

Posteriormente foram introduzidos fatores de ponderação, para permitir a aplicação de outros tipos de estacas. Para a resistência de ponta, foi introduzido o coeficiente α e para o atrito lateral o coeficiente β, descrito respectivamente na tabela 3, para cada tipo de estaca e para cada tipo de solo:

Solo	Estaca Pré-fabricada		Escavada sem lama (broca)		Escavada com lama		Hélice contínua		Estaca Raiz		Microestaca	
	α	β	α	β	α	β	α	β	α	β	α	β
Argila	1,00	1,00	0,85	0,80	0,85	0,90	0,30	1,00	0,85	1,50	1,00	3,00
Silte	1,00	1,00	0,60	0,65	0,60	0,75	0,30	1,00	0,60	1,50	1,00	3,00
Areia	1,00	1,00	0,50	0,50	0,50	0,60	0,30	1,00	0,50	1,50	1,00	3,00

TABELA 3: COEFICIENTES α E β PARA AS ESTACAS.
FONTE: Schnaid (2000).

Estes coeficientes de ponderação são utilizados para o cálculo da resistência de ponta e a atrito lateral, que são descritas pelas respectivas equações (ALONSO, 1991):

$$R_p = \alpha \times q_{p,ult} \times S_P \quad (6)$$

$$R_l = \beta \times q_{l,ult} \times S_l \quad (7)$$

Onde:

- R_p: Resistência de ponta;
- α: Coeficiente de ponderação referente à resistência de ponta (Tabela 3);
- $q_{p,ult}$: Resistência de ponta unitária;
- S_P: Área da base;
- R_l: Atrito lateral;
- β: Coeficiente de ponderação referente ao atrito lateral (Tabela 3);
- $q_{l,ult}$: Atrito lateral unitário;
- S_l: Perímetro do fuste.

Com a resistência de ponta e o atrito lateral, pode-se calcular a carga de ruptura, definida pela seguinte equação (ALONSO, 1991):

$$P_r = R_p + R_l \quad (8)$$

Onde:

- P_r: Carga de ruptura.

Com a carga de ruptura calculada, pode-se obter a capacidade de carga, que é a carga que realmente age no elemento de fundação, pelas equações (ALONSO, 1991):

$$\bar{P} = {}^{P_r}/_{2} \quad (9)$$

$$\bar{P} = \left({}^{R_P}/_{4}\right) + \left({}^{R_l}/_{1,3}\right) \quad (10)$$

Onde:

- $\bar{P}$: Capacidade de carga.

Para a capacidade de carga deve-se adotar sempre o menor valor obtido nas equações anteriores.

2.6 Aspectos gerais sobre as estacas empregadas no método semi-empírico de Décourt-Quaresma

A seguir serão apresentadas as principais características e parâmetros que devem ser considerados para cada tipo de estaca.

2.6.1 Estaca Pré-fabricada de concreto

De todos os materiais da construção, o concreto e o aço, podem ser considerados os melhores para produção de estacas pré-moldadas, devido ao controle de qualidade que pode-se ter, tanto para confecção quanto para cravação (HACHICH, 1998).

Este tipo de estaca, é constituída de segmentos de concreto pré-fabricado e introduzida no terreno por golpes de martelo de gravidade, de explosão, hidráulico ou vibratório (NBR 6122, 2010).

As estacas pré-fabricadas podem ser executadas por prensagem, por vibração ou por percussão, métodos estes, descritos abaixo (HACHICH, 1998):

- Prensagem: para este tipo de cravação são utilizados macacos hidráulicos que reagem contra uma plataforma com sobrecarga ou até mesmo na própria estrutura;
- Vibração: utiliza-se um martelo provido de garras para a fixação na estaca, com massas excêntricas, que ao girarem rapidamente produzem uma vibração de alta frequência que é transmitida à estaca;
- Percussão: este é o processo mais utilizado, para isso utilizam-se pilões de queda livre ou automáticos, sendo esta mais eficiente, devido a continuidade dos golpes na estaca.

A grande vantagem das estacas deste tipo é obtenção da qualidade desejável do concreto e no fato de que os agentes encontrados no solo, não terão efeito na pega e na cura deste concreto (VELLOSO; LOPES, 2011).

As estacas pré-moldadas de concreto podem ter a carga variando entre 200 kN a 1500 kN. A seguir estão listadas em quais situações não se recomendam o uso deste tipo de estaca (ALONSO, 1983):

- Terrenos com presença de rochas ou de pedregulhos;

- Terrenos que a cota da ponta da estaca seja variável, devido ao perfil geotécnico do local, não conseguindo manter um comprimento constante;
- Deve-se avaliar as construções vizinhas, pela vibração causada pela execução, pode causar danos as construções vizinhas;
- Solos com N_{spt} elevado (>25).



FIGURA 4 – ESTACAS PRÉ-FABRICADAS DE CONCRETO.
FONTE: PDI – Pernambuco Desenvolvimento Industrial.

2.6.2 Estaca escavada com lama

A estaca escavada com lama é moldada in loco, sendo que a estabilidade da parede de perfuração é assegurada pelo uso de fluido estabilizante ou água quando for revestimento metálico (NBR 6122, 2010).

Esta técnica surgiu em torno de 1952, as estacas são executadas nas mais diversas condições de terreno, com comprimentos que ultrapassam 50m e seções transversais de até 2,5m de diâmetro (VELLOSO; LOPES, 2011).

Essas estacas são executadas com o emprego de lama bentonítica, para suporte das escavações e concretagens submersas. Esta lama é composta por uma mistura de água e bentonita, gerando assim a lama bentonítica, que tem como características a rápida formação, em superfícies porosas, de uma película impermeável, o que estabiliza a parede da perfuração (HACHICH, 1998).

O processo executivo das estacas escavadas com lama bentonítica, compreende as seguintes fases (HACHICH, 1998):

- Escavação e preenchimento simultâneo da estaca com lama bentonítica previamente preparada;
- Posicionamento da armadura antecipadamente montada, dentro da escavação cheia de lama;
- Lançamento do concreto, através de um tubo denominado tremonha. A concretagem é feita de baixo para cima, sendo o concreto mais denso que a lama, a lama é expulsa e bombeada de volta para depósitos.

As estacas escavadas com lama bentonítica são utilizadas para cargas elevadas, ou seja, acima de 1500 kN. Estas estacas não causam vibração em obras vizinhas, porém necessitam de um grande espaço para a instalação dos equipamentos, visto que são de grande porte (ALONSO, 1983).



FIGURA 5 – ESTACA ESCAVADA COM A PRESENÇA DE LAMA BENTONÍTICA.
FONTE: LAN – Luiz A. Naresi Jr – Estacas escavadas com auxílio de lama.

2.6.3 Estaca Hélice contínua

A estaca hélice contínua refere-se a uma estaca de concreto moldada no local, executada pela introdução por rotação de um trado helicoidal contínuo no terreno e a injeção do concreto dá-se pela própria haste central do trado simultaneamente com a sua retirada, sendo que a armadura é introduzida após a estaca concretada (NBR 6122, 2010).

O trado da hélice contínua tem grande comprimento e é composto de chapas metálicas em formato espiralado que se desenvolvem em torno do tubo central. A extremidade inferior do trado possui uma tapa que impede a entrada de solo no tubo central além de garras para facilitar o corte do terreno. Os equipamentos usuais executam estacas de 30cm a 100 cm de diâmetro e comprimentos de até 30 metros de profundidade (VELLOSO; LOPES, 2011).

A perfuração dá-se pela introdução do trado no terreno, por meio de movimentos de rotação transmitidos por motores hidráulicos, até a cota estabelecida em projeto sem que ocorra a retirada do trado do terreno. Através do tubo central é bombeado concreto continuamente até acima da cota de arrasamento, simultaneamente com a retirada do trado, sem girar ou girando lentamente no sentido da perfuração. A pressão no concreto deve garantir que ela preencha todos os vazios deixados pela extração da hélice. A “gaiola” de armadura deve ser introduzida após a concretagem que pode ser imersa no concreto manualmente por operários, com auxílio de um peso ou até mesmo com auxílio de um vibrador. Na extremidade inferior da gaiola de armadura deve se ter barras ligeiramente inclinadas, formando uma espécie de cone para facilitar a introdução do concreto além de espaçadores tipo rolete (VELLOSO; LOPES, 2011).

A execução das estacas pode ser monitorada eletronicamente através de um computador ligado a sensores instalados na máquina, com isso temos alguns elementos monitorados, como a inclinação, o torque, o volume de concreto entre outros (VELLOSO; LOPES, 2011).

Este tipo de estaca não se deve executar com espaçamento inferior a cinco diâmetros em intervalo inferior a 12 horas. Este espaçamento refere-se à estaca de maior diâmetro (NBR 6122, 2010).

As cargas admissíveis para este elemento de estacas está entre 300 kN à 5000 kN (GEOSONDA, 2012).

Alguns parâmetros devem ser considerados para a escolha deste elemento de fundação, abaixo estão os itens a serem verificados (HACHICH, 1998):

- As áreas devem ser planas, devido ao porte dos equipamentos de cravação;

- Devido a uma grande produtividade, deve-se ter uma central de concreto próxima ao equipamento;
- Limitação de comprimento das estacas e das armaduras, os equipamentos disponíveis permitem executar estacas com no máximo 24m;
- O sistema pode ser empregado em qualquer tipo de solo, exceto em solos que contenham matacões, rochas e pedregulhos.



FIGURA 6 – ESTACAS HÉLICE CONTÍNUA.
FONTE: Fundamenta – Engenharia de Fundações.

2.6.4 Estaca raiz

A estaca-raiz é definida pelo tipo de execução, que pode ser por perfuração rotativa ou roto-percussiva ou por uso de revestimento integral no trecho do solo. Realiza-se um preenchimento com argamassa cimento areia, após a colocação da armadura em todo comprimento, a argamassa é adensada com o auxílio de pressão geralmente dada por ar comprimido (NBR 6122, 2010).

As estacas-raiz foram utilizadas inicialmente para contenção de encostas, posteriormente utilizadas em reforço de fundações e por último em fundações normais (VELLOSO; LOPES, 2011).

Estas estacas possuem características que permitem sua utilização em casos específicos, onde os demais tipos não podem ser empregados, pois elas não

produzem choques nem vibrações, é possível executa-las através de obstáculos como rocha e peças de concreto, equipamentos de pequeno porte, possibilitando acesso a ambientes restritos e podem ser executadas com qualquer inclinação (VELLOSO; LOPES, 2011).

Para a perfuração utiliza-se normalmente o processo rotativo, com circulação de água ou lama bentonítica. Caso encontrado material mais resistente a perfuração pode-se prosseguir com uma coroa diamantada. Após terminada a perfuração introduz-se a armadura de aço que pode ser em formato de “gaiola” ou simplesmente uma única barra. A concretagem é realizada com uma argamassa composta por areia e cimento e é bombeada por um tubo até a ponta da estaca, à medida que a argamassa sobe pelo tubo de revestimento o mesmo é retirado e são aplicados golpes de ar comprimido que adensam a argamassa e promovem o contato com o solo (VELLOSO; LOPES, 2011).



FIGURA 7 – ESTACA RAIZ.
FONTE: Tec Sonda – Tecnologia em sondagem LTDA.

As cargas para a estaca raiz variam entre 350 kN e 1300 kN (GEOFIX, 2014).

As estacas raiz podem ser executadas nos seguintes locais (SÃO PAULO, E. P. U, 2003):

- Áreas de dimensões reduzidas;
- Capacidade de execução em qualquer tipo de solo;

- Em muitos casos, como reforços de fundação, devido ao fato de que os equipamentos são de pequeno porte;
- Em locais de difícil acesso, por exemplo, onde já existem construções;
- Execução da estaca com inclinações variáveis de 0º a 90º.

2.6.5 Microestaca (injetada sob alta pressão)

A NBR 6122:2010, descreve a Microestaca da seguinte maneira: são aquelas moldadas in loco, armadas, executadas através de perfuração rotativa ou roto-percussiva e injetada com calda de cimento por meio de um tubo com válvulas, denominadas manchete.

O processo executivo pode ser definido por três etapas (VELLOSO; LOPES, 2011):

- Perfuração: como já descrito acima, usa-se o processo rotativo, com circulação de água ou lama bentonítica. No caso de solos com argilas moles e areias fofas, utiliza-se um tubo de revestimento provisório;
- Armadura: pode ser constituída de uma gaiola de barras de aço ou por um tubo de aço com válvulas manchetes, essas válvulas são furos no tubo de aço, tapados com borracha, através delas que a calda de cimento é injetada sob pressão. No caso de ser gaiola, o tubo de aço é colocado dentro da gaiola;
- Injeção: pode-se processar quantas vezes for necessário até chegar na pressão desejável. Após estas injeções, ocorre o enchimento do tubo de injeção com argamassa ou calda de cimento, obtendo-se um fuste irregular, semelhante ao bulbo de um tirante.

Para minimizar o custo das microestacas, pode-se trocar os tubos metálicos por tubo de PVC rígido, mas neste caso é obrigatório o uso de armadura, visto que este tubo de PVC não tem função estrutural (HACHICH, 1998).

Nas microestacas, a armadura é envolvida por argamassa ou por calda de cimento, porém nunca com concreto (HACHICH, 1998).



FIGURA 8 – MICROESTACA.
FONTE: Geo-Rumo – Tecnologia de Fundações.

2.6.6 Estaca escavada sem lama (Broca)

Essas estacas são em grande parte executadas com trados manuais com diâmetros entre 20cm e 40cm, e por trados mecânicos que podem ter diâmetros maiores. A estaca broca é escavada com trado manual e empregada em situações em que a base fica acima do lençol d'agua ou em situações onde pode-se assegurar que o furo estará seco antes da concretagem (VELLOSO; LOPES, 2011).

A execução desta estaca é relativamente simples, uma vez atingida a profundidade prevista, deve-se fazer a limpeza do fundo, com a remoção de material proveniente da escavação. A concretagem é feita com concreto, que é lançado de superfície do terreno com auxílio de um funil (VELLOSO; LOPES, 2011).

A armadura utilizada geralmente utiliza conjuntos de ferros longitudinais, amarrados com estribos em espiral. Esta ligação atende à ligação com o bloco de coroamento e, se necessário, pode ter o comprimento da estaca e resistir a outros esforços da estrutura (VELLOSO; LOPES, 2011).

A estaca broca pode ser adotada para as seguintes situações (ALONSO, 1983):

- Pequenas cargas, variando entre 50 kN a 100 kN;
- Devem estar sempre acima do nível de água;

- O diâmetro varia entre 15 cm e 25 cm;
- Comprimento em torno de 3 m (para trado manual).



FIGURA 9 – ESTACA BROCA COM TRADO MANUAL.
FONTE: Pró-Solo – Sondagens e Fundações LTDA.



FIGURA 10 – ESTACA BROCA COM TRADO MECÂNICO.
FONTE: AGM – Geotécnica LTDA.

2.7 Estaqueamento

Estaqueamento pode ser definido como qualquer conjunto de duas ou mais estacas que irão receber as cargas oriundas da estrutura e realizar a transmissão desta para o solo (ALONSO, 1983).

Para o cálculo do número de estacas, quando a carga atuante coincide com o centro do estaqueamento, o número de estacas pode ser calculado pela seguinte equação (ALONSO, 1983):

$$N = \frac{P_{PILAR}}{\bar{P}} \tag{11}$$

Onde:

- N: Número de estacas constituintes na estrutura de ligação;
- P_{PILAR}: Carga do pilar;
- $\bar{P}$: Capacidade de carga.

A locação das estacas deve ser feita ao redor do centro das cargas das mesmas, e sempre que possível, para se obter um menor consumo de material, deve-se levar em consideração alguns blocos padronizados (ALONSO, 1983).

Outra situação que pode ocorrer, são pilares com cargas verticais e esforços de momento fletor. Para o cálculo deste estaqueamento, usualmente é utilizado o método da superposição, método este que consiste em determinar a carga em cada estaca, somando-se separadamente os efeitos provenientes das cargas verticais e dos momentos fletores. A Figura 11 demonstra um estaqueamento submetido a uma carga vertical e ao esforço de momento fletor, em ambas as direções (ALONSO, 1983).

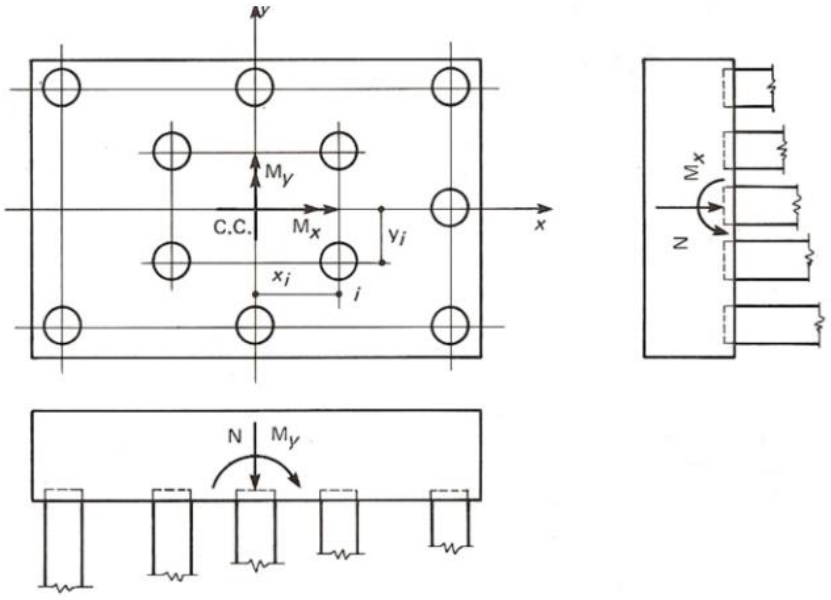


FIGURA 11 – ESTAQUEAMENTO COM CARGAS VERTICAIS E MOMENTOS FLETORES EM AMBAS AS DIREÇÕES.
FONTE: Alonso (1983).

Para a realização do método citado acima, os eixos principais, x e y, devem ser os eixos principais de inércia e que as estacas sejam do mesmo comprimento, do mesmo tipo e do mesmo diâmetro. A carga atuante em cada estaca, é o resultado da seguinte equação (ALONSO, 1983):

$$P_i = \frac{P_{PILAR}}{N} \pm \frac{M_y \times x_i}{\sum x_i^2} \pm \frac{M_x \times y_i}{\sum y_i^2} \quad (12)$$

Onde:

- P_i: Carga atuante na estaca i;
- P_{PILAR}: Carga vertical do pilar;
- N: Número de estacas do estaqueamento;
- M_y: Momento fletor transmitido pelo pilar na direção y;
- M_x: Momento fletor transmitido pelo pilar na direção x;
- x_i: coordenada da estaca i, na direção x;
- y_i: coordenada da estaca i, na direção y;

Para estaqueamentos sujeitos aos esforços de momento fletor, é solucionado através de tentativas, adotando uma quantidade de estacas e suas respectivas coordenadas, pode-se calcular a carga atuante nas mesmas. Este estaqueamento somente será aceito se a carga nas estacas for menor ou igual a carga de trabalho (ALONSO, 1983).

2.8 Visual Basic

John Kemeny e Thomas Kurtz desevolveram uma linguagem de programação para estudantes, chamada BASIC. Esta linguagem foi criada para pessoas iniciantes em programação de computadores, tornando mais fácil a aprendizagem da linguagem computacional. Com o passar dos anos e com novas tecnologias no mercado, foram criadas novas versões. Com o surgimento do ambiente Windows, criado pela Microsoft, foi apresentado em 1991 a versão do Visual Basic, esta versão passou a fazer uso de recursos visuais, além de possuir uma estrutura orientada a eventos. O Visual Basic ganhou diversas atualizações, até que no ano de 2001 a Microsoft introduziu o Visual Basic com suporte a plataforma de programação .NET, onde a estrutura operacional da linguagem foi remodelada e passou a ser referenciada também como VB.NET. Em 2012 foi apresentado o

pacote Visual Studio, onde tem a capacidade de integrar diversas linguagens de programação e a cada ano que se passa novas versões são lançadas com atualizações e melhorias (MANZANO, 2014).

2.9 Banco de dados – XML

O XML (eXtensible Markup Language), que significa linguagem de marcação extensível criado em 1996 pelo World Wide Web Consortium, o W3C, foi baseada em um padrão já existente, o SGML (Standard Generalized Markup Language. O XML foi criado para ser usado na internet e também pode ser utilizado para armazenamento de dados (DAUM; MERTEN, 2002).

O XML armazena dados em uma estrutura em árvores. A vantagem mais importante do uso do XML para armazenamento é a simplicidade, estas estruturas em árvores contém uma maneira simples e eficaz de organizar dados. Outra vantagem é que este armazenamento de dados suporta o conjunto de caracteres Unicode (BRODGEN; MINNICK, 2002).

Unicode é a maneira oficial de implementar o UCS (Universal Chareter Set), que é o conjunto de caracteres universal definido pela ISO (International Standards Organization) (BRODGEN; MINNICK, 2002).

2.10 Revisão Bibliográfica

Neste subcapítulo são descritos alguns trabalhos correlatos que merecem destaque e que foram analisados durante o desenvolvimento do presente trabalho.

Foi realizada uma dissertação de mestrado com o tema “otimização dos custos dos projetos de estaqueamento através de uma planilha de cálculo eletrônica de fundações profundas”, na qual é buscada uma solução ótima, em função do menor custo total, que leva em conta o custo de escavação/cravação por metro linear e o custo de concreto armado para blocos de fundação (CARVALHO, 2014).

Para determinação da capacidade de carga nas estacas, foram utilizados os métodos de Aoki-Velloso modificado por Monteiro e Décourt-Quaresma para estacas hélice contínua, Aoki-Velloso modificado por Monteiro, Décourt-Quaresma, Brasfond e Fundesp para estacas raiz e Aoki-Velloso modificado por Monteiro e

Décourt-Quaresma para estacas pré-moldadas, adotando entre os métodos o valor mais conservador. Para as armaduras das estacas moldadas in loco foi utilizado o método de Miche, já as estacas pré-moldadas foram consideradas conforme a capacidade retirada de tabelas. A planilha calcula o esforço atuante nas estacas considerando blocos de fundações previamente padronizados e regulares, estes blocos variam de uma a seis estacas, os blocos são rígidos e os carregamento atuantes são: momento fletor, forças verticais e horizontais determinadas pelo usuário. Para finalizar, a autora realiza um estudo de caso, onde compara o estaqueamento de uma obra real com o dimensionamento realizado pela planilha e observa que, para esta obra, haveria uma redução de 35,84% em relação ao projeto original, neste caso usou-se estacas pré-moldadas centrifugadas de 50 e 70cm, conforme apresentado em sua dissertação de mestrado. (CARVALHO, 2014).

Em um artigo desenvolvido para a REEC – Revista Eletrônica de Engenharia Civil, intitulado “Rotina computacional para a previsão da capacidade de carga em estacas” os autores apresentaram uma rotina computacional para o cálculo da capacidade de carga em estacas, comparando os resultados obtidos por meio de provas de carga. Esta planilha foi desenvolvida utilizando o “Excel” com a implementação de macros e linguagem, de programação VBA. Para o cálculo foram utilizados os métodos semiempiricos propostos por Aoki-Velloso, Décourt-Quaresma e Teixeira (FERREIRA; DELELIBERA; SILVA, 2014).

Na planilha o usuário deve informar o tipo de estaca, diâmetro, a cota da ponta da estaca e por fim a sondagem, logo após entrar com estes dados, o usuário escolhe o método para cálculo. A rotina irá apresentar os valores da capacidade de carga das estacas de forma automática, podendo analisar todos os parâmetros calculados para cada método, e por fim é apresentado um resumo dos resultados obtidos e também apresenta um gráfico que representa a capacidade de carga ao longo de todo o fuste. Esta planilha não apresenta a melhor solução com relação a custo por metro de escavação/cravação de cada estaca, como descrito acima, ela apenas calcula a capacidade de carga das estacas, podendo não ser a melhor opção entre muitas (FERREIRA; DELELIBERA; SILVA, 2014).

3 PROCEDIMENTOS METODOLÓGICOS

Para atender aos objetivos de dimensionamento de fundações profundas e selecionar a solução mais econômica, foi desenvolvida uma ferramenta denominada *CALFES – Calculo de Fundações em Estacas*.

Para tanto, serão descritas as principais funções do software gerado, nos itens subsequentes.

3.1 Fluxograma do *CALFES*

O funcionamento do software, está ilustrado no fluxograma apresentado na figura 12, que representa o processamento dos dados por meio de funções.

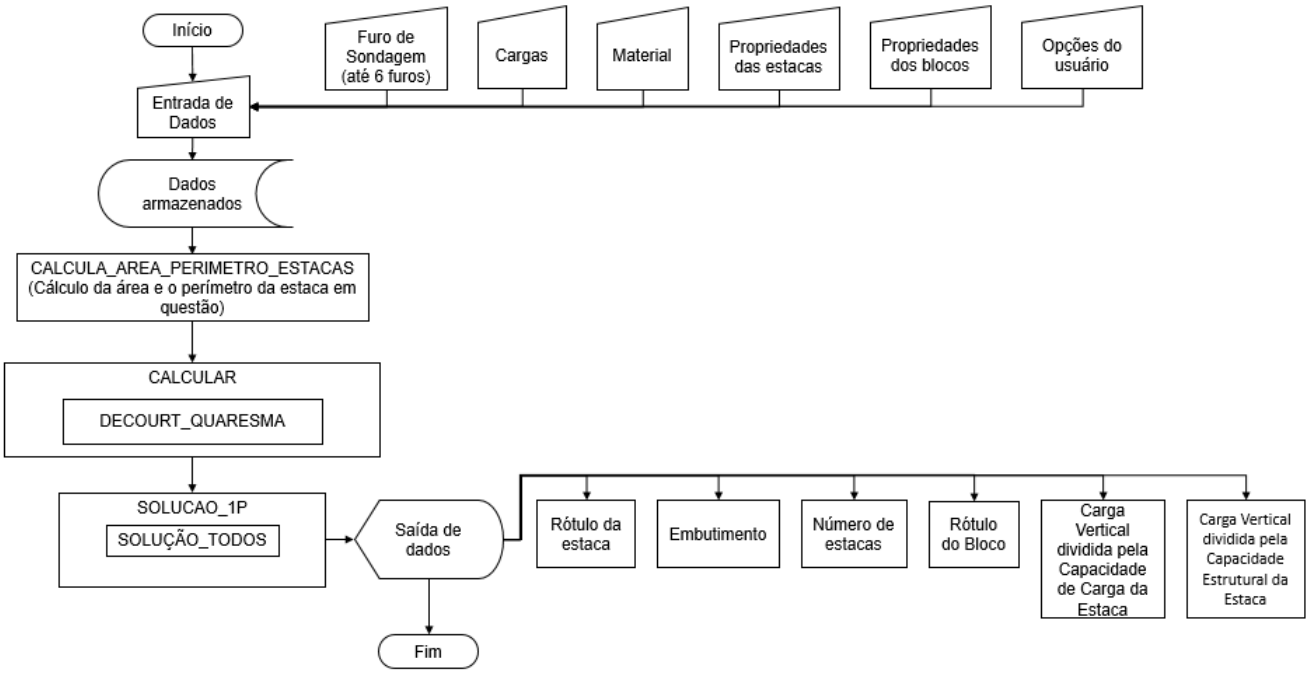


FIGURA 12 – FLUXOGRAMA GERAL DO FUNCIONAMENTO DO SOFTWARE.
FONTE: Os Autores (2016).

As funções integrantes do CALFES, ilustradas na figura 12, serão descritas nos subtítulos subsequentes.

3.2 Entrada de dados

A entrada de dados é a parte inicial do software, onde cabe ao usuário definir os parâmetros a serem utilizados para o dimensionamento das estacas, do estudo em questão. Todas as entradas de dados, estão apresentadas na figura 13.

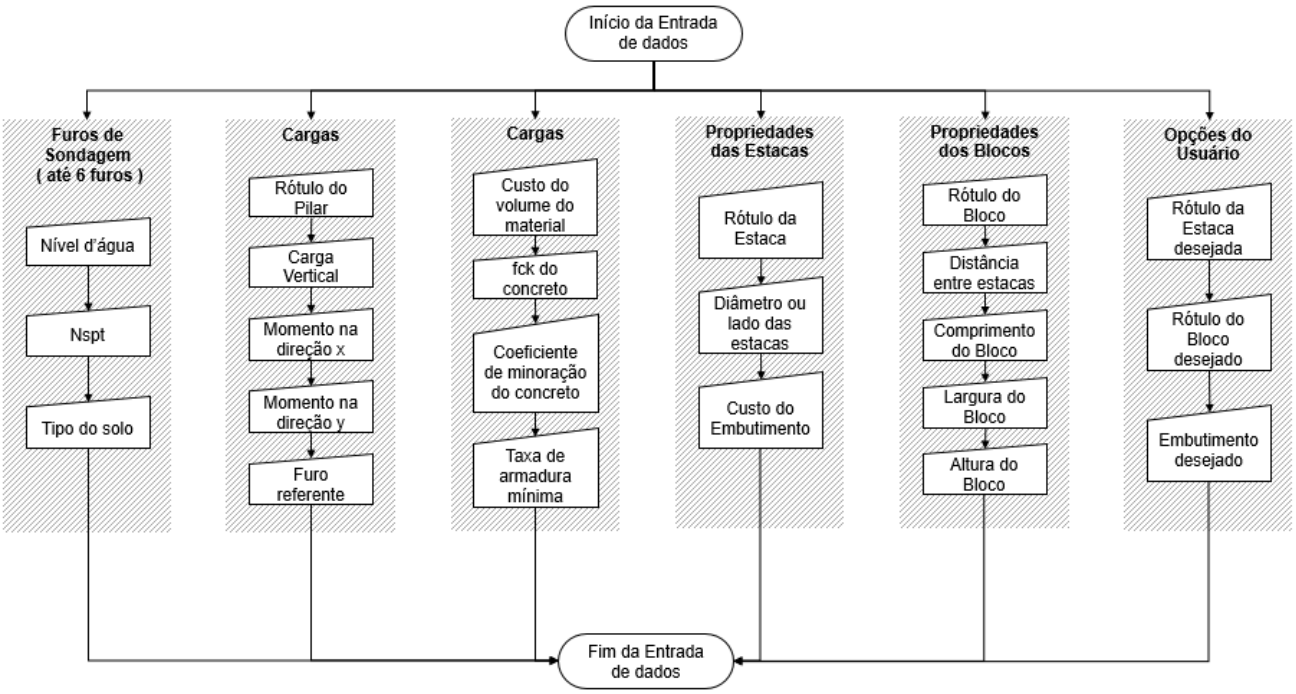


FIGURA 13 – FLUXOGRAMA DA ENTRADA DE DADOS.
FONTE: Os Autores (2016).

O usuário deverá informar ao software, os dados retirados dos perfis geotécnicos levantados, a partir de ensaios SPT (Standard Penetration Test), os dados de nível d'água, os índices de resistência a penetração, o N_{spt} e indicar para cada metro prospectado os tipos de solo. Para esta entrada de dados o usuário poderá informar até 6 perfis de sondagens.

Além dos dados pertinentes às sondagens, o usuário deverá informar o rótulo do pilar, ou seja, como quer identificar o pilar estudado e deverá informar as cargas solicitantes por pilar. Estas cargas poderão ser: de compressão e momento fletor, em ambas as direções. Caso não contenha esforços de momento fletor, o usuário não precisará preenchê-las. Nesta etapa o usuário deverá relacionar este pilar a um devido furo de sondagem.

As próximas entradas de dados são referentes ao material utilizado nas estacas. O usuário deverá informar qual é o custo, por metro, do volume de material,

bem como a resistência caracteristica a compressão do concreto, o $f_{ck.}$ Ainda nesta etapa, o usuário terá que informar, segundo a tabela 1 descrita neste trabalho, o coeficiente de minoração do concreto e a taxa mínima de armadura.

Logo após a inserção destes dados, deverá ser inserido no software os dados pertinentes as propriedades das estacas escolhidas para o dimensionamento. Em primeiro lugar, o usuário definirá qual o tipo de estaca possível para o estudo que terá de ser analisado pelo software, deverá informar a seção transversal da mesma e o custo de embutimento por metro desta estaca.

E se submetida ao esforço de momento fletor, o usuário deverá informar o rótulo do bloco, ou seja, qual bloco será analisado pelo software e as propriedades geométricas do bloco de coroamento. Para estas propriedades deverá ser informado a distância entre eixos das estacas, que não poderá ser menor que 2,5 diâmetros, o comprimento do bloco, a largura e a altura.

Por fim o usuário terá entradas de dados opcionais, dentre estas estão o rótulo da estaca, onde poderá definir para todas as cargas inseridas, um único tipo de estaca, o embutimento desejado para a solução e a quantidade de estacas desejada. Após realizada a entrada de dados, os mesmos serão armazenados.

3.3 Dados armazenados

Após a inserção de todos os dados necessários para o processamento do software, ambos são armazenados em um banco de dados XML. Este banco de dados armazena as informações inseridas em formato de tabelas, e correlaciona cada linha e cada coluna com sua respectiva função. Funções estas que serão apresentadas nos subtítulos subsequentes.

3.4 Função: CALCULA_AREA_PERIMETRO_ESTACAS

Nesta função ocorre o cálculo pertinente às áreas e os perímetros relativos às seções transversais, fornecidas pelo usuário.

A função calcula as áreas para estacas circulares vazadas, estacas circulares com seção transversal cheia e estacas quadradas, e seus respectivos perímetros.

O funcionamento desta função está representado pela figura 14.

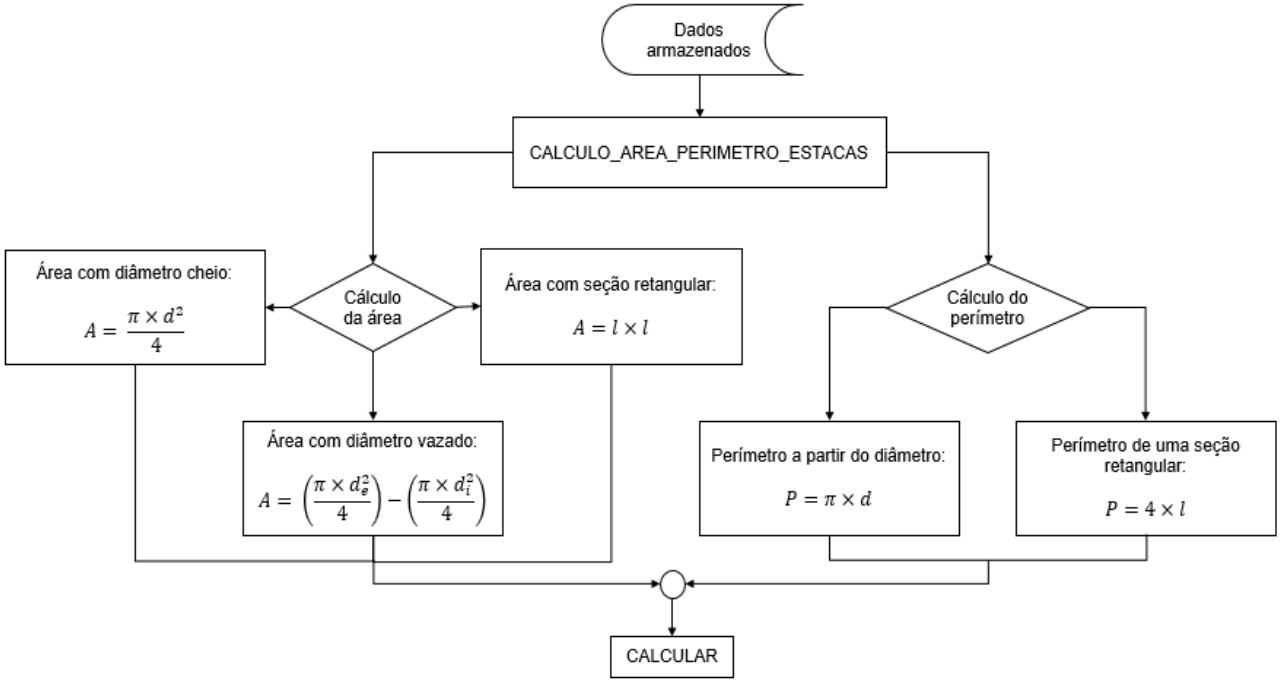


FIGURA 14 – FLUXOGRAMA DA FUNÇÃO: CALCULA_AREA_PERIMETRO_ESTACAS.
FONTE: Os Autores (2016).

3.5 Função: CALCULAR

O funcionamento da função está apresentado na figura 15.

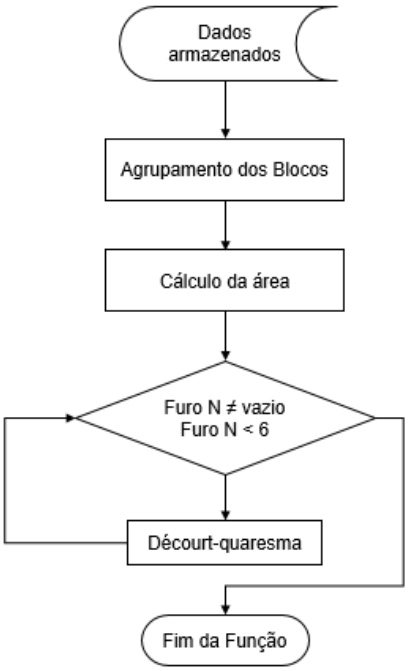


FIGURA 15 – FLUXOGRAMA DA FUNÇÃO: CALCULAR.
FONTE: Os Autores (2016).

Após os armazenamentos dos dados, a primeira das etapas da função é agrupar todos os dados referentes aos blocos em uma datagridview, onde também já é calculado o volume respectivo a cada bloco e armazenado esse valor.

Em seguida é iniciada a função Cálculo da Área e armazenado na memória RAM estes valores. Para iniciar a função Decourt_Quaresma o software realiza uma verificação, se a entrada de dados referente ao furo de sondagem está vazia, caso negativa, a partir do índice do furo referente inicia-se a função Decourt-Quaresma para aquele furo de sondagem representado pelo índice, caso positivo verifica-se o próximo furo, até terminar a verificação dos seis furos possíveis.

3.6 Função: DECOURT_QUARESMA

Com os dados armazenados, será realizado pelo software a verificação da capacidade de carga, pela função DECOURT_QUARESMA.

Se determinada uma profundidade, o software fará o cálculo da capacidade de carga por metro, até o último metro prospectado. Outra hipótese que pode ser adotada é deixar com que o software calcule, para a carga desejada, a melhor profundidade possível. Para isso a função calcula, a resistência de ponta, o atrito lateral, a carga de ruptura e pôr fim a capacidade de carga.

A resistência de ponta será calculada pelas equações e pelos parâmetros expostos na fundamentação. Para este cálculo, serão utilizados o coeficiente α, que varia em função do tipo do solo e do tipo da estaca, e será utilizado a área da seção transversal calculada pela função descrita no item 3.4.

O atrito lateral calculará a capacidade de carga resistida pela estaca ao longo do fuste, por metro linear. Para este cálculo não são considerados valores de N_{spt} maiores que 50 e valores menores que 3, para os valores fora deste intervalo, são considerados os limites do intervalo, ou seja, o 3 e o 50. O N_{spt} utilizado para o cálculo da resistência de ponta será desconsiderado para este cálculo. Para este cálculo, serão utilizados o coeficiente β, que varia em função do tipo do solo e do tipo da estaca, e será utilizado a área da seção transversal calculada pela função descrita no item 3.4.

O funcionamento desta função está representado pela figura 16.

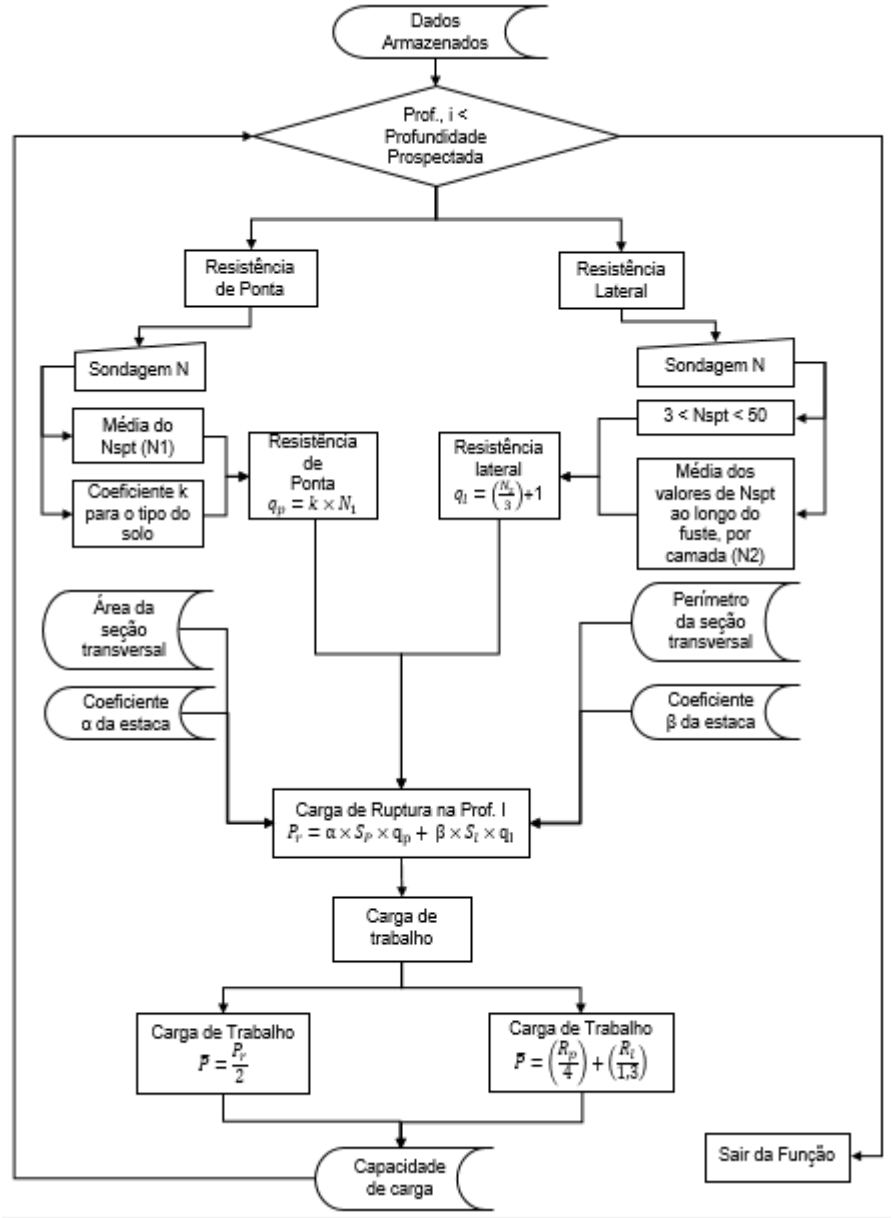


FIGURA 16 – FLUXOGRAMA DE FUNCIONAMENTO DA FUNÇÃO: DÉCOURT-QUARESMA.
FONTE: Os Autores (2016).

A carga de ruptura será calculada pela simples soma entre a resistência de ponta e o atrito lateral. Com a carga de ruptura calculada, o software realiza duas verificações para determinar a carga de trabalho, que também pode ser chamada de capacidade de carga. Realizadas as verificações, o software compara as duas cargas de ruptura calculadas e seleciona a menor dentre elas. Os resultados obtidos por esta função são armazenados no banco de dados. Todas as considerações para

a determinação desta capacidade de carga de trabalho estão de acordo com o método semi-empírico de Décourt-Quaresma.

3.7 Função: SOLUCAO_1P

O funcionamento da função está apresentado nas figuras 17 e 18.

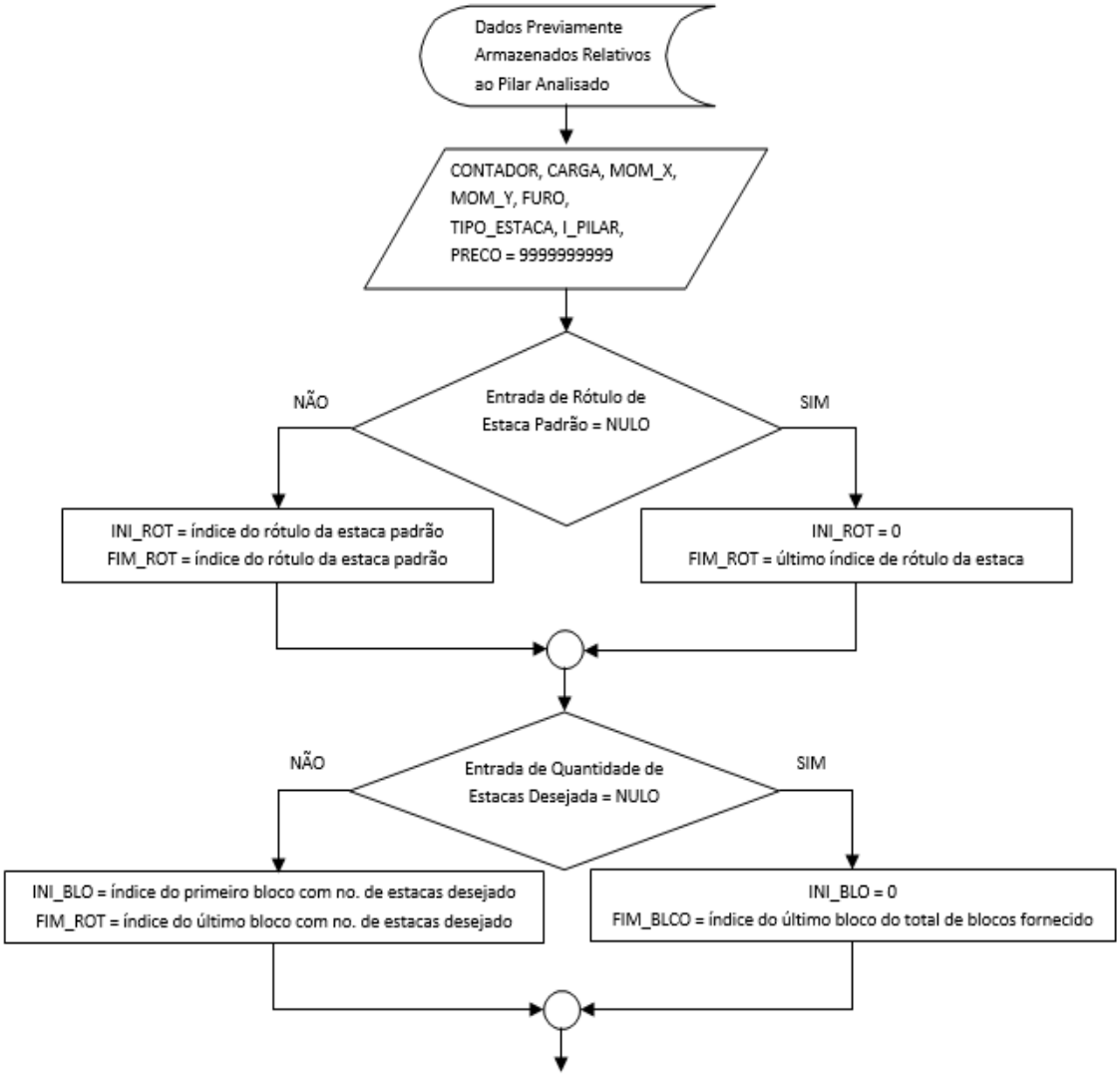


FIGURA 17 – FLUXOGRAMA DA FUNÇÃO: SOLUCAO_1P – PARTE 01.
FONTE: Os Autores (2016).

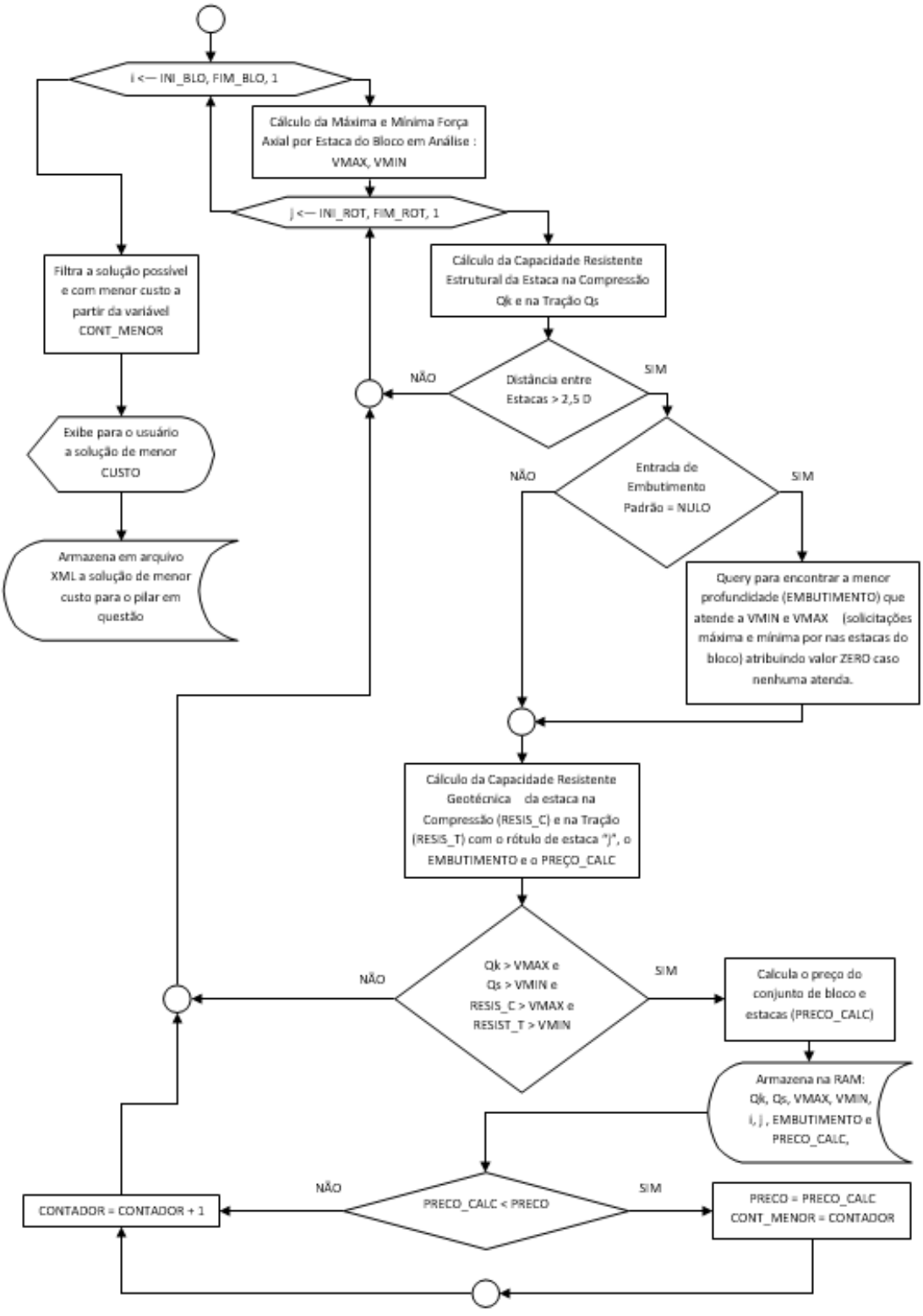


FIGURA 18 – FLUXOGRAMA DA FUNÇÃO: SOLUCAO_1P – PARTE 02.
FONTE: Os Autores (2016).

A função SOLUCAO_1P é realizada apenas para um pilar em questão, uma das entradas desta função é o valor referente a linha. A partir dos dados armazenados, são selecionados alguns, como o momento na direção x, na direção

y, carga de compressão, furo referente a linha do pilar e o tipo de estaca em questão. Também é armazenada numa varável local um valor referente ao preço.

A primeira condicional do programa é verificar se a entrada de dados opcional que determina o rótulo da estaca é nula, caso sim o software armazena na variável ini_rot o índice inicial e na variável fim_rot o índice final de todos as estacas preenchidas pelo usuário, caso não é armazenado apenas na variável ini_rot e fim_rot o índice da linha que referencia o rótulo dado.

A segunda condicional é verificar se a entrada de dados referente a quantidade de estacas é nula, caso sim, de mesmo modo que acontece com o rótulo da estaca, é armazenado o índice inicial na variável ini_rot e final na variável fim_blo referente a todos os blocos de estacas dado pelo usuário, caso não, são armazenados apenas os índices das linhas que representam a quantidade de blocos que possuem o número de estacas dado.

A partir dos índices já selecionados (ini_blo a fim_blo), é realizado uma estrutura de laço, dentro dela calcula-se a máxima e a mínima força axial na estaca.

Abaixo desta estrutura de laço é percorrida outra estrutura de laço com os índices das estacas (ini_rot a fim_rot), calcula-se então a capacidade estrutural da estaca a compressão e a capacidade de carga estrutural a tração.

Então é verificado para cada estaca se o diâmetro multiplicado por 2,5 é menor que o espaçamento entre estacas, se não, verifica-se para a próxima estaca, se sim, verifica –se se na entrada de dados o valor da profundidade é nulo, caso afirmação positiva, é executada uma query que procura uma profundidade no qual a capacidade resistente do solo atende a máxima e a mínima solicitação axial, caso não encontre nenhum embutimento é atribuído valor zero. Caso seja arbitrado um valor de entrada para a profundidade desejada ou encontrado um valor na query acima é calculado o valor da capacidade geotécnica resistente a tração e a capacidade geotécnica resistente a compressão.

É então verificado se máxima solicitação axial de compressão é menor que a capacidade estrutural a compressão e menor que a capacidade geotécnica resistente, de mesma forma, para a máxima solicitação axial de tração, caso não seja atendido as verificações acima, faz-se a verificação para a próxima estaca, caso a verificação seja satisfeita, calcula –se o custo total de cada solução possível

(preço do embutimento das estacas mais volume de concreto do bloco e das estacas multiplicado pelo preço do volume do concreto).

Satisfeita todas estas condições, e armazenadas na memória RAM as informações acima (Capacidade estrutural de compressão e de tração, solicitações máximas e mínimas de forças axiais, índices referentes aos blocos e as estacas, embutimentos referentes e o preço total calculado).

É realizada uma verificação se o preço calculado é menor que o preço adotado na variável local inicialmente, se sim é armazenado nesta variável local o valor do preço calculado e verificado para a próxima solução e armazenado o índice da solução na variável CONT_MENOR, se não, mantém-se o valor e verificado a próxima solução. Acaba-se o ciclo do laço referente as estacas, índices ini_rot a fim_rot.

Filtra-se a solução de menor custo a partir do valor armazenado na variável CONT_MENOR, é exibido então para o usuário o menor custo para uma solução possível, resistência solicitante/resistente, resistência solicitante/estrutural, rótulo adotado, embutimento mínimo possível e quantidade de estacas. Então armazena em xml a opção de menor custo para o pilar em questão.

3.8 Função: SOLUCAO_TODOS

O funcionamento desta função está representado pela figura 19.

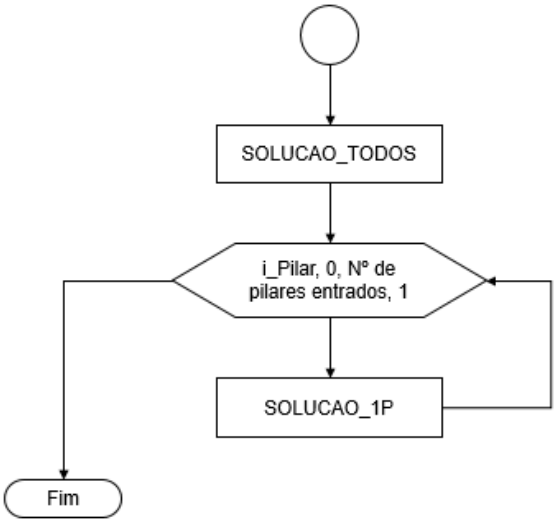


FIGURA 19 – FLUXOGRAMA DA FUNÇÃO: SOLUCAO_TODOS.
FONTE: Os Autores (2016).

É admitido como o índice inicial para os pilares o índice zero e índice final a quantidade de pilares na entrada de dados. Uma estrutura de laço percorre a função SOLUCAO_1P e então próximo índice.

3.9 Saída de dados

A saída de dados está apresentada na figura 20.

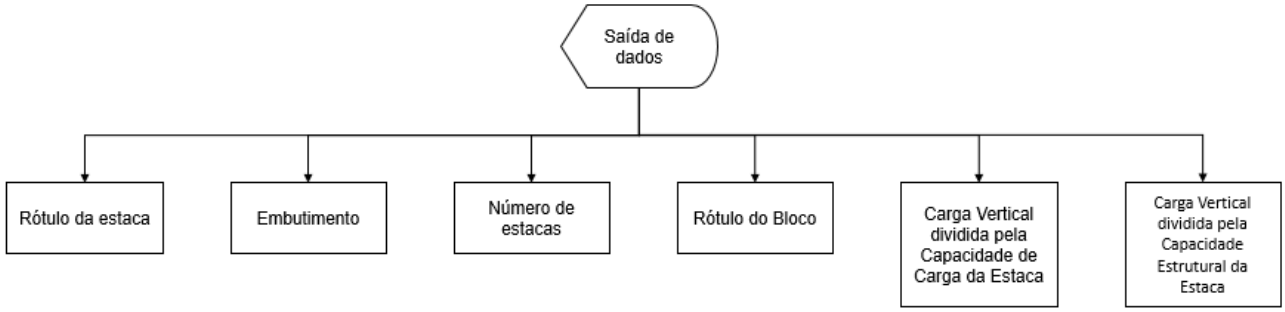


FIGURA 20 – FLUXOGRAMA DA SAÍDA DE DADOS.
FONTE: Os Autores (2016).

Obtidos todos os resultados, pelas funções descritas, o usuário terá a possibilidade de variar os dados para obter outros resultados.

O usuário poderá definir um embutimento igual para todas as estacas, da obra em questão, e verificar a quantidade e as seções transversais, pertinentes aos tipos de estaca escolhido. Poderá optar também, em definir a mesma seção transversal para todas as estacas, e verificar o embutimento de todas as soluções correlatas e avaliar a quantidade de estacas. Terá também a opção de variar os rótulos das estacas, ou seja, os tipos de estacas consideradas para o método de Décourt-Quaresma, e verificar as novas seções transversais e seus respectivos embutimentos.

No fim do processamento, poderá ser salvo um arquivo, em formato .xml e se necessário, fazer alterações em algum dado, o usuário poderá abri-lo novamente e realizar a mudança e o novo processamento. O usuário poderá gerar um relatório com o resumo das estacas estudadas.

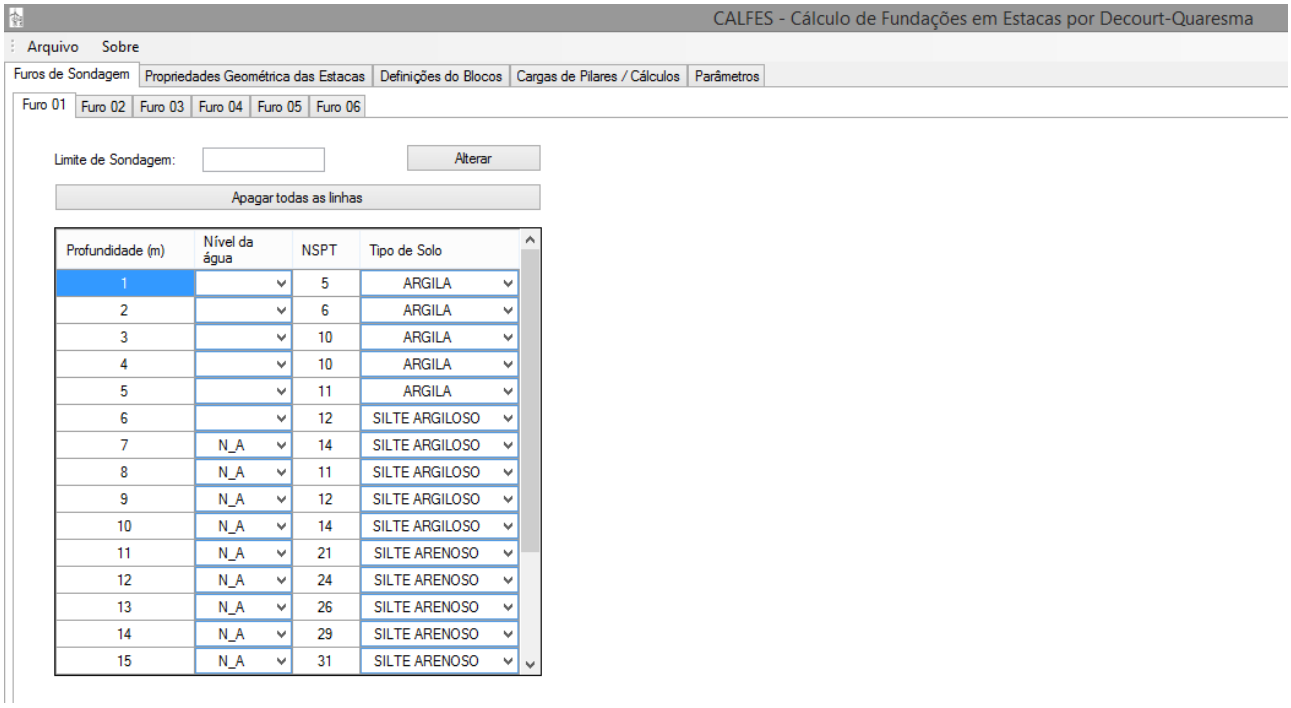
3.10 Validação do Software

Para validação do software, será criado um exemplo, que comtemplará todos os procedimentos citados acima. Os resultados deste exemplo estão descritos no próximo capítulo.

4 RESULTADOS

Para se obter resultados para comparação e validação com o software, foi criado um exemplo para tal propósito. Os dados utilizados no exemplo estão apresentados no Apêndice B.

As figuras 21 e 22, fazem parte da entrada de dados do programa, nesta etapa o usuário dá entrada no perfil geotécnico do solo, com dados obtidos pelo ensaio SPT (Standard Penetration Test), sendo estes dados: o nível da água, o Nspt e o tipo do solo.



Profundidade (m)	Nível da água	NSPT	Tipo de Solo
1		5	ARGILA
2		6	ARGILA
3		10	ARGILA
4		10	ARGILA
5		11	ARGILA
6		12	SILTE ARGILOSO
7	N_A	14	SILTE ARGILOSO
8	N_A	11	SILTE ARGILOSO
9	N_A	12	SILTE ARGILOSO
10	N_A	14	SILTE ARGILOSO
11	N_A	21	SILTE ARENOSO
12	N_A	24	SILTE ARENOSO
13	N_A	26	SILTE ARENOSO
14	N_A	29	SILTE ARENOSO
15	N_A	31	SILTE ARENOSO

FIGURA 21 – FURO DE SONDAGEM 01 – PARTE 01.
FONTE: Os Autores (2016).

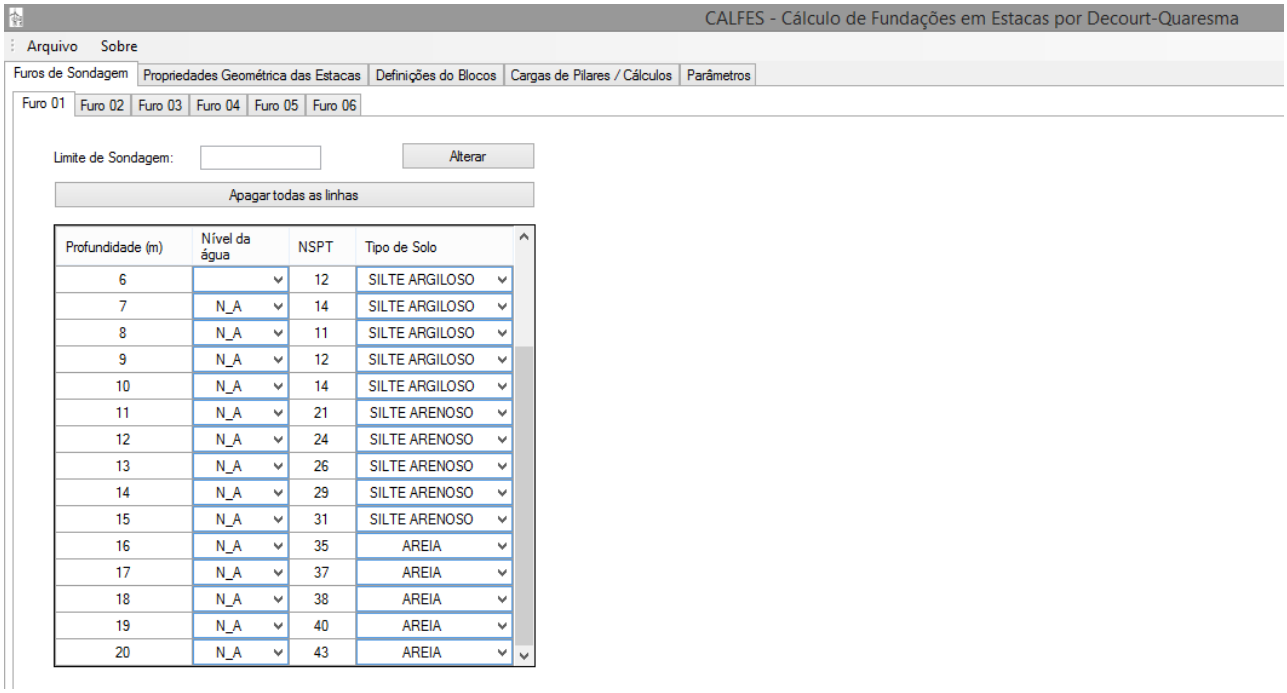


FIGURA 22 – FURO DE SONDAGEM 01 – PARTE 02.
FONTE: Os Autores (2016).

As propriedades geométricas das estacas são definidas conforme a figura 23, onde o usuário define o tipo de estaca a ser calculada, a seção transversal e o preço do embutimento por metro, além de definir o nome para a mesma.

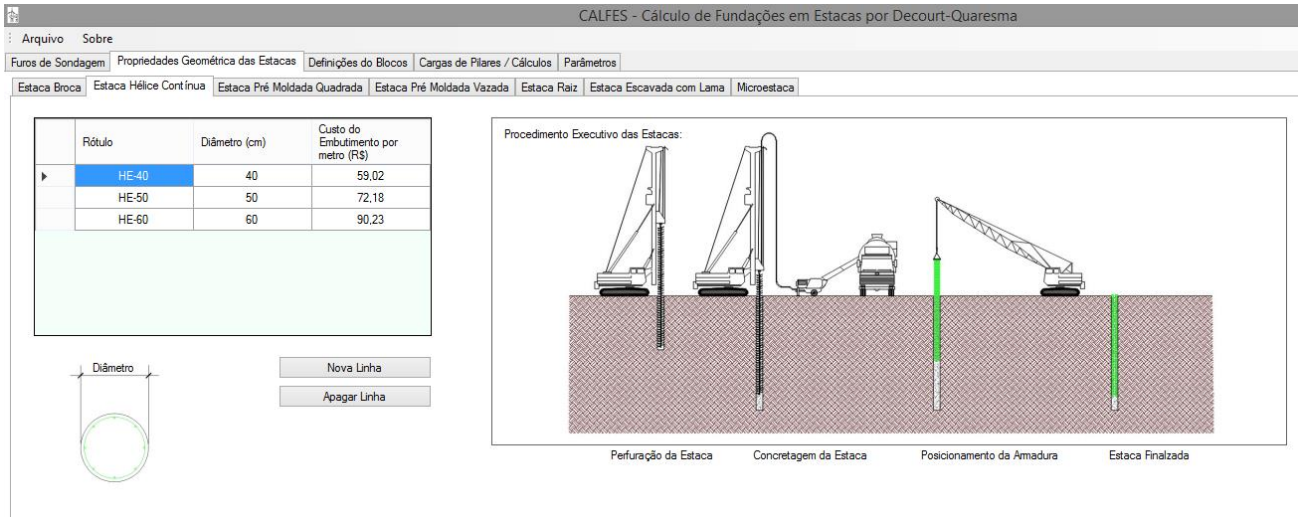


FIGURA 23 – PROPRIEDADES GEOMÉTRICA DAS ESTACAS HÉLICE CONTÍNUA.
FONTE: Os Autores (2016).

Com possibilidade de escolher blocos de coroamento de uma até nove estacas o usuário define quais blocos podem ser exequíveis no estudo em questão.

Para este caso foram definidos os blocos com uma estaca, com duas estacas e com quatro estacas. Atentando que o bloco com quatro estacas foi definido de duas maneiras diferentes, ou seja, com dois conjuntos de dimensões (comprimento, largura e altura). As definições dos blocos estão representadas nas figuras 24, 25 e 26.

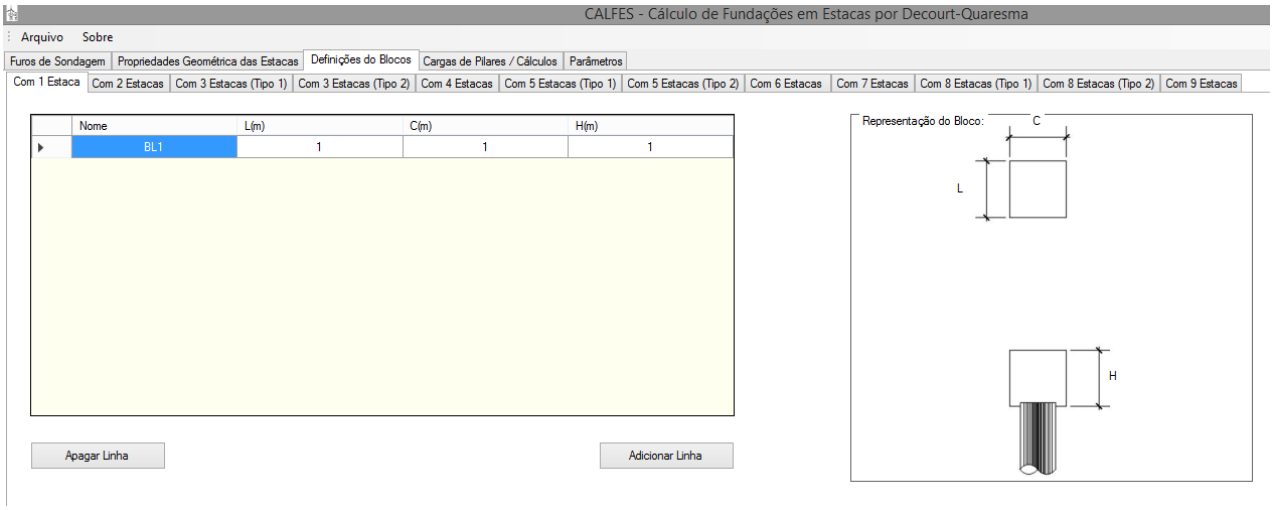


FIGURA 24 – ENTRADA DE DADOS PARA BLOCO COM 1 ESTACA.
FONTE: Os Autores (2016).

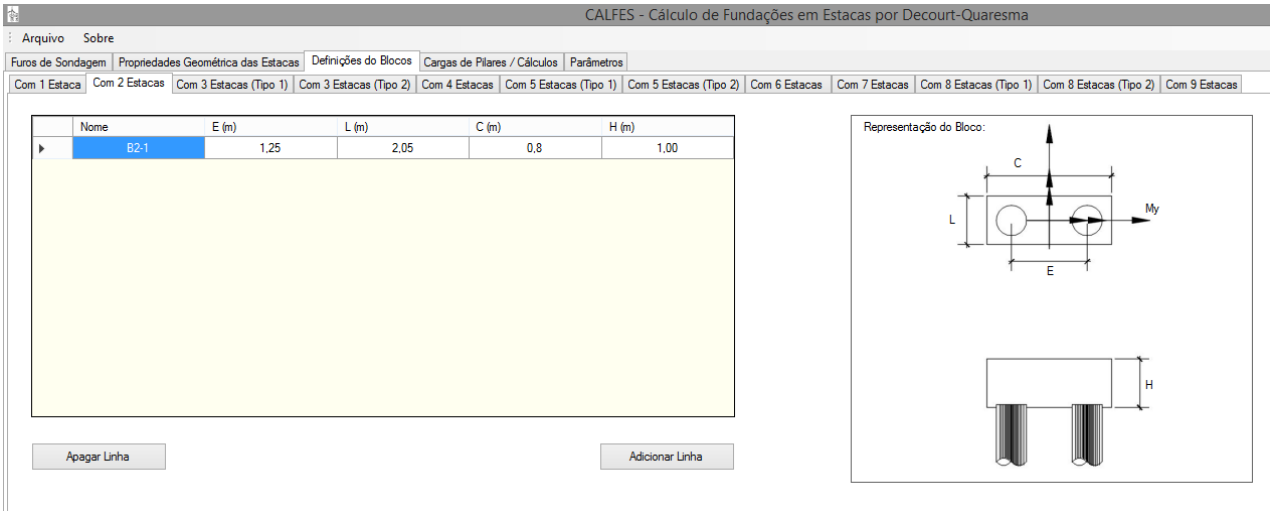


FIGURA 25 – ENTRADA DE DADOS PARA BLOCO COM 2 ESTACA.
FONTE: Os Autores (2016).

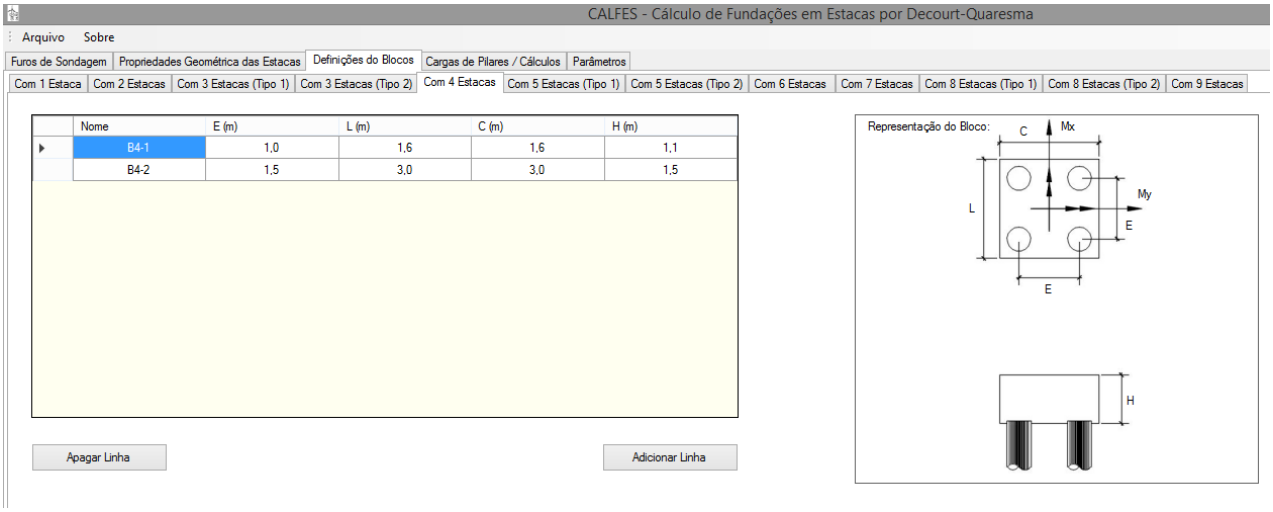


FIGURA 26 – ENTRADA DE DADOS PARA BLOCO COM 4 ESTACA.
FONTE: Os Autores (2016).

Os parâmetros utilizados para o cálculo são default. Na figura 27 estão representados os dados pertinentes à capacidade estrutural e a capacidade geotécnica de cada tipo de estaca.

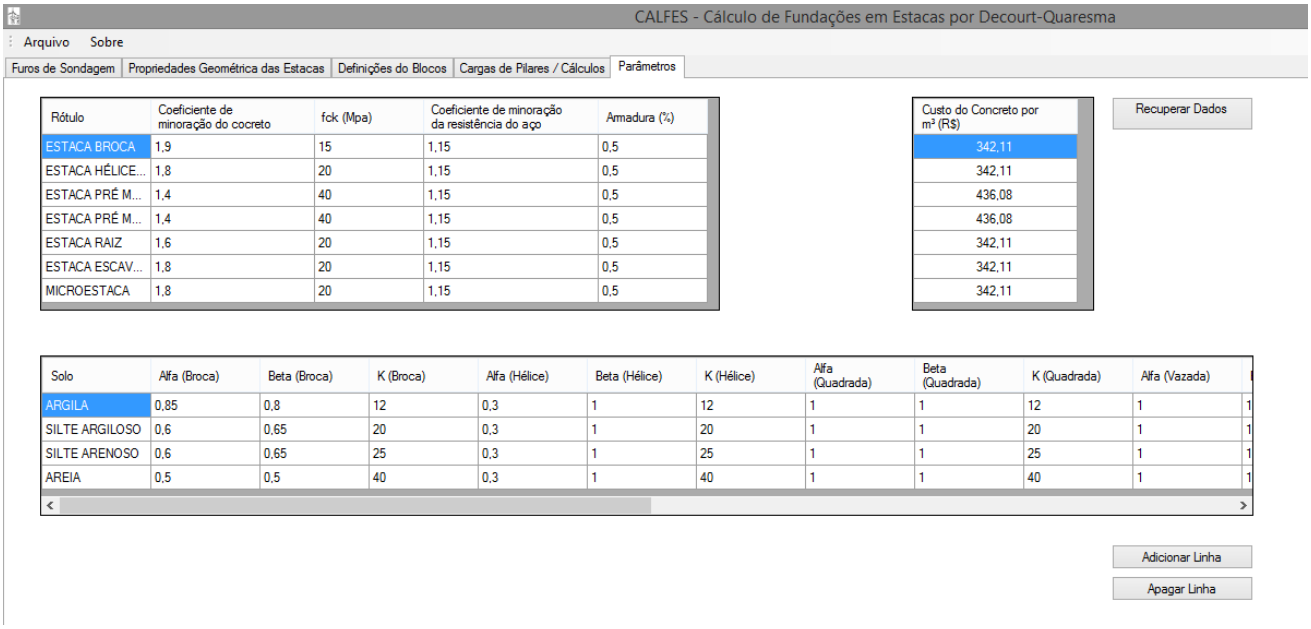


FIGURA 27 – PARÂMETROS PARA O DIMENSIONAMENTO.
FONTE: Os Autores (2016).

A figura 28 representa as entradas de dados referentes à identificação do pilar, as solicitações (Carga axial de compressão e momento fletor em ambas as direções) e o furo de sondagem referente ao pilar. Neste caso o usuário não definiu nenhum rótulo, embutimento ou a quantidade de estaca desejada, deixando a cargo

do software calcular os mesmos. O exemplo em questão considera tanto a resistência de ponta, quanto a resistência lateral.

Ainda na figura 28, estão representados os resultados obtidos pelo software, estes são: o rótulo calculado, o embutimento calculado, a quantidade de estacas calculadas e o bloco referente, a razão entre a carga solicitante e a carga resistente, a razão entre a carga solicitante e a capacidade estrutural, preço total por pilar e o custo total da solução.

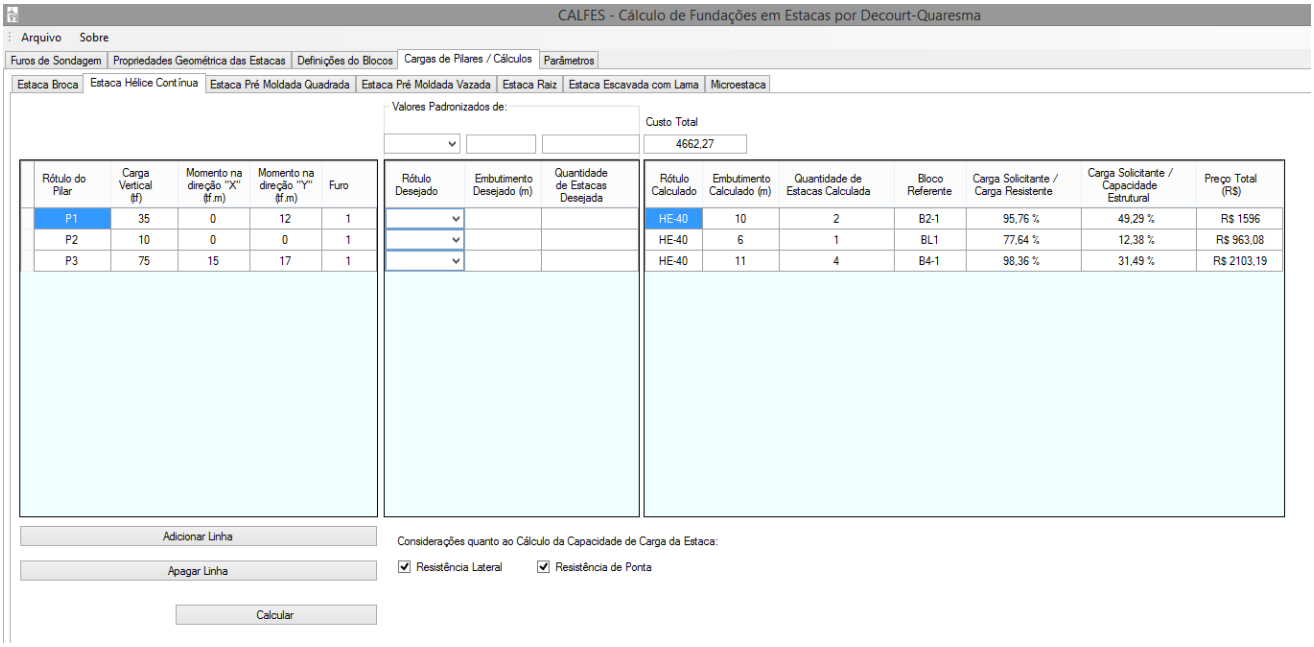


FIGURA 28 – CARGAS DOS PILARES E CALCULOS PARA ESTACA HÉLICE CONTÍNUA.
FONTE: Os Autores (2016).

Na sequência serão apresentados, para título de comparação, os resultados referentes ao exemplo criado com aplicação de cálculos manuais e aplicação do software, respectivamente, respeitando todas as premissas de cálculo descritas na fundamentação teórica deste trabalho.

A tabela 4 exibe os resultados via cálculo manual e a figura 29 exibe o relatório gerado pelo software para fins de comparação.

	Rótulo do Bloco	Volume do Bloco (m^3)	Rótulo da Estaca	Área da seção transversal da estaca (m^2)	Embuti-mento mínimo (m)	Custo do Concreto (R$/$m^3$)	Custo do Embutimento (R$/m)	Custo Total (R$)	Menor Custo (R$)
PILAR P1	BL2	1,64	HE-40	0,13	10,00	342,11	59,02	1596,00	1596,00
	BL2	1,64	HE-50	0,20	8,00	342,11	72,18	1685,88	
	BL4-1	2,82	HE-40	0,13	7,00	342,11	59,02	1689,21	
	BL4-2	13,50	HE-40	0,13	7,00	342,11	59,02	5342,95	
	BL4-2	13,50	HE-50	0,20	6,00	342,11	72,18	5462,10	
	BL4-2	13,50	HE-60	0,28	6,00	342,11	90,23	5734,61	
PILAR P2	BL-1	1,00	HE-40	0,13	6,00	342,11	59,02	963,08	963,08
	BL-1	1,00	HE-50	0,20	5,00	342,11	72,18	1045,12	
	BL-1	1,00	HE-60	0,28	4,00	342,11	90,23	1086,19	
	BL2	1,64	HE-40	0,13	4,00	342,11	59,02	975,04	
	BL2	1,64	HE-50	0,20	3,00	342,11	72,18	982,87	
	BL4-1	2,82	HE-40	0,13	3,00	342,11	59,02	1275,23	
	BL4-2	13,50	HE-40	0,13	5,00	342,11	59,02	5135,96	
	BL4-2	13,50	HE-50	0,20	5,00	342,11	72,18	5321,50	
	BL4-2	13,50	HE-60	0,28	4,00	342,11	90,23	5362,57	
PILAR P3	BL4-1	2,82	HE-40	0,13	11,00	342,11	59,02	2103,19	2103,19
	BL4-2	13,50	HE-40	0,13	12,00	342,11	59,02	5860,42	
	BL4-2	13,50	HE-50	0,20	10,00	342,11	72,18	6024,51	
	BL4-2	13,50	HE-60	0,28	9,00	342,11	90,23	6292,67	

TABELA 4 – RESULTADOS A PARTIR DE CÁLCULOS MANUAIS.
FONTE: Os Autores (2016).

CALFES - Cálculo de Fundações em Estacas por Decourt-Quaresma

Resultados de Dimensionamento

TIPO DE ESTACA: HELICE

Pilares	CARGAS	ESTACA	EMBUT.	QTD.	BLOCO	FSD/FRD	FSD/CEST	CUSTO
P1	35	HE-40	10	2	B2-1	95,76 %	49,29 %	R$ 1596
P2	10	HE-40	6	1	BL1	77,64 %	12,38 %	R$ 963,08
P3	75	HE-40	11	4	B4-1	98,36 %	31,49 %	R$ 2103,19

CUSTO TOTAL: 4662,27

FIGURA 29 – RELATÓRIO COM RESULTADOS GERADOS PELO SOFTWARE.
FONTE: Alonso (1983).

5 ÁNALISE DOS RESULTADOS

Com base nos resultados apresentados no capítulo 4 é possível realizar uma análise comparativa das respostas obtidas através dos cálculos via software e as respostas obtidas através dos cálculos manuais.

Pode-se observar que comparando os valores obtidos pelas duas soluções, não ocorreu nenhuma diferença entre os mesmos comprovando que o software replica exatamente o procedimento de cálculo descrito na fundamentação teórica.

Nota-se que cada pilar possui uma quantidade de soluções possíveis, com diferentes custos e que o software é capaz de selecionar a de menor custo dentre estas.

Um dos resultados apresentados para o usuário é o embutimento calculado, que é a menor profundidade possível para que a capacidade resistente geotécnica da estaca seja menor que a solicitação axial. É exibida para o usuário uma razão entre a solicitação axial e a capacidade resistente geotécnica de modo a balizar o aproveitamento do material empregado na solução.

Vale ressaltar que algumas análises não são realizadas, como por exemplo, quando o espaçamento entre as estacas é menor que 2,5 vezes o diâmetro da mesma, o que acontece com a estaca de 60 centímetros de diâmetro e os blocos B2 e BL4-1. Além disso, quando um pilar possui momentos na direção X e direção Y o software só considera blocos que possuem capacidade de resistir a ambos momentos, como por exemplo, os blocos BL4-1 e BL4-2, assim sendo, blocos como o Bloco 1, que só admite cargas axiais e não momento, é desprezado no processo de cálculo.

Outro resultado apresentado pelo software é o rótulo calculado da estaca, neste caso para todos os pilares a solução de menor custo apresentada foi à estaca HE-40. Para selecionar a estaca em questão o software verifica se a capacidade estrutural é menor que a força axial solicitante, seja ela de compressão ou tração, caso verifique, o programa exibe para o usuário uma razão da capacidade estrutural da estaca com a solicitação axial.

Nas verificações comprovou-se que nenhuma solicitação axial foi maior que a capacidade estrutural das estacas selecionadas. Nos casos em que a solicitação foi maior, o software funciona buscando uma estaca de seção transversal maior ou um bloco com maior número de estacas sendo a resistência geotécnica do solo a determinante para o cálculo do embutimento.

6 CONSIDERAÇÕES FINAIS

Por meio do desenvolvimento deste trabalho, foi desenvolvido um código computacional que originou o software *CALFES – Calculo de Fundações em Estacas*. O software pré-dimensiona fundações profundas em estacas, baseado no método semi-empírico de Décourt-Quaresma. O programa possibilita trabalhar com todas as estacas possíveis de serem dimensionadas com o método.

Quanto aos resultados obtidos pelo software o mesmo se mostra satisfatório, proporcionando ao usuário acadêmico um auxílio na elaboração do pré-dimensionamento de estacas.

O método de Décourt-Quaresma foi o escolhido, pois este é embasado nos resultados de ensaios de sondagem à percussão (SPT) que é o ensaio mais comumente utilizado, além de ser o método mais difundido no Brasil. A NBR 6122:2010 foi determinante no que se diz respeito à capacidade estrutural da estaca e sua armadura mínima, conforme descrito no subtítulo 2.4.

Como vimos anteriormente, os cálculos manuais para a determinação da capacidade de carga e busca da solução mais econômica representam um processo lento que o software consegue efetuar em poucos segundos. Dessa forma conclui-se que o objetivo proposto foi atingido pela composição de uma ferramenta computacional capaz de processar o pré-dimensionamento de soluções geotécnicas em estacas com rapidez e eficiência permitindo a comparação entre várias possíveis soluções.

Tendo em vista que o software atende aos objetivos propostos, sugere-se que futuras complementações sejam feitas, como por exemplo, a verificação estrutural do bloco via método das bielas e a implementação de outros métodos de dimensionamento de fundações profundas em estacas, como por exemplo, o Método de Aoki-Velloso e ainda a possibilidade de pré-dimensionar tubulões.

REFERÊNCIAS

ALONSO, U. R. **Previsao e controle das fundaçoes**. [s.l.] EDGARD BLUCHER, [s.d.]

ALONSO, U. R. **Exercícios de fundações**. [s.l.] E. Blucher, [s.d.]

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 14724**: Informação e documentação – Apresentação de trabalhos acadêmicos. Rio de Janeiro: ABNT, 2005.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6118**: Projeto de estruturas de concreto - Procedimento. Rio de Janeiro: ABNT, 2014.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6122**: Projeto e execução de fundações. Rio de Janeiro: ABNT, 2010.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6484**: Solo – Sondagens de simples reconhecimento com SPT – Método de ensaio. Rio de Janeiro: ABNT, 2001.

BRODGEN, B.; MINICK, C. **Guia de Desenvolvedor JAVA – Desenvolvendo E-Commerce com JAVA, XML e JSP**. MAKRON Books, São Paulo, 2002.

CARVALHO, F. V. M. de C. **Otimização dos custos dos projetos de estaqueamento através de uma planilha de cálculo eletrônica de fundações porfundas**. 2014. Escola Politécnica da Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2014.

CARVALHO, R. C.; DE FIGUEIREDO FILHO, J. R. **Cálculo e detalhamento de estruturas usuais de concreto armado: segundo a nbr 6118:2003**. [s.l.] EDUFSCAR, 2004.

DAUM, B.; MERTEN, U. **Arquitetura de sistemas com XML**. Editora Campos, Rio de Janeiro, 2002.

DE ALENCAR VELLOSO, D.; DE REZENDE LOPES, F. **Fundações: critérios de projeto, investigação do subsolo, fundações superficiais, fundações profundas**. [s.l.] Oficina de Textos, 2011.

E. SIMONS, N.; K. MENZIES, B. **Introdução à engenharia de fundações**. [s.l.] Editora Interciência, 1981.

FERREIRA, T. R.; DELALIBERA, R. G.; SILVA, W. A. da S. REEC - Revista Eletrônica de Engenharia Civil. **Rotina computacional para a previsão da capacidade de carga em estacas**, v. 8, n. 3, p. 38–50, jul. 2014.

GEOFIX. **4º Curso de Engenharia aplicada às obras de fundações e contenções**. FAAP – Fundação Armando Alvares Penteado, São Paulo, 2014.

GEOSONDA, S. A. **Hélice Contínua Monitorada**. São Paulo, 2016.

HACHICH, W. C. **Fundações: teoria e prática**. [s.l.] Pini, 1998.

MANZANO, J. A. N. G. **Estudo Dirigido de Microsoft Visual Basic Express 2013**. [s.l.] Editora Érica, 2014.

SÃO PAULO, E. P. U. **Tecnologia da construção de edifícios – Fundações.** Universidade de São Paulo, Revisão em Fevereiro de 2003.

SCHNAID, F. **Ensaios de campo e suas aplicações à engenharia de fundações**. [s.l.] Oficina de Textos, 2000.

APÊNDICE A (CÓDIGO COMPUTACIONAL)

A seguir será apresentado o código computacional realizado para a criação do software.

```
Imports System.IO
Imports System.Xml
Imports System.Xml.Linq
Imports System.Math
Imports System.Data
Imports iTextSharp.text.pdf
Imports iTextSharp.text
Imports System.Console


Public Class FormPrincipal

    Dim TEXTO As String
    Dim endereco As String

    Private Sub TX(TXT)
        TEXTO += TXT & vbNewLine
    End Sub

    Private Sub LOG()
        Dim FILE_NAME As String = "LOG.txt"
        Dim objWriter As New System.IO.StreamWriter(FILE_NAME)
        objWriter.Write(TEXTO)
        objWriter.Close()
    End Sub
    'MANIPULAÇÃO DE ARQUIVOS
    Private Sub NOVO_ARQUIVO(sender As Object, e As EventArgs) Handles
NovoToolStripMenuItem.Click
        endereco = ""
        BANCO_DE_DADOS.Clear()
        FORM1_LOAD(sender, e)
    End Sub
    Private Sub SAIR(sender As Object, e As EventArgs) Handles
SairToolStripMenuItem1.Click
        Me.Close()
    End Sub
    Private Sub SALVAR_COMO(sender As Object, e As EventArgs) Handles
SalvarComoToolStripMenuItem.Click
        Try
            Me.SaveFileDialog1.Filter = "XML (*.xml)|*.xml"
            Me.SaveFileDialog1.ShowDialog()
            endereco = SaveFileDialog1.FileName
            BANCO_DE_DADOS.WriteXml(endereco)
        Catch
            MsgBox("Erro ao salvar o arquivo!!! Tente outra vez.")
        End Try
    End Sub
    Private Sub SALVAR(sender As Object, e As EventArgs) Handles
SalvarToolStripMenuItem.Click
        Try
            MsgBox("Dados salvos no arquivo : " & endereco)
            BANCO_DE_DADOS.WriteXml(endereco)
```

```
        Catch
            SALVAR_COMO(sender, e)
        End Try
    End Sub
    Private Sub ABRIR(sender As Object, e As EventArgs) Handles
AbrirToolStripMenuItem.Click
        Me.SaveFileDialog1.Filter = "XML (*.xml)|*.xml"
        Me.OpenFileDialog1.ShowDialog()
        endereco = OpenFileDialog1.FileName
        Dim xmlFile As XmlReader
        xmlFile = XmlReader.Create(endereco, New XmlReaderSettings())
        BANCO_DE_DADOS.Clear()
        BANCO_DE_DADOS.ReadXml(xmlFile)
        LIBERA_CALCULO()
    End Sub
    'FUNÇÕES DE FORMULÁRIO
    Private Sub FORM1_LOAD(sender As Object, e As EventArgs) Handles MyBase.Load
        ALTERA_DGVCOMBOBOX_ROTULO_FURO(FURO_01DataGridView)
        ALTERA_DGVCOMBOBOX_ROTULO_FURO(FURO_02DataGridView)
        ALTERA_DGVCOMBOBOX_ROTULO_FURO(FURO_03DataGridView)
        ALTERA_DGVCOMBOBOX_ROTULO_FURO(FURO_04DataGridView)
        ALTERA_DGVCOMBOBOX_ROTULO_FURO(FURO_05DataGridView)
        ALTERA_DGVCOMBOBOX_ROTULO_FURO(FURO_06DataGridView)
        LIBERA_CALCULO()
        TAB_GERAL.TabPages.Remove(tab_dimen)
        TAB_GERAL.TabPages.Remove(tab_obra)
        RECUPERAR_DADOS_PADRAO(sender, e)
    End Sub
    Private Sub ALTERA_DGVCOMBOBOX_ROTULO_FURO(FURO As Object)
        Dim alfa As DataGridViewComboBoxColumn
        alfa = FURO.Columns.Item(1)
        alfa.Items.Clear()
        alfa.Items.Add("N_A")
        alfa.Items.Add("")
    End Sub
    Private Sub LIBERA_CALCULO()
        If EST_BROCADataGridView.RowCount = 0 Then
            ENT_BROCADataGridView.Visible = False
            SOL_BROCADataGridView.Visible = False
            LABEL_01.Visible = True
        Else
            ENT_BROCADataGridView.Visible = True
            SOL_BROCADataGridView.Visible = True
            LABEL_01.Visible = False
        End If
        If EST_HELICEDataGridView.RowCount = 0 Then
            ENT_HELICEDataGridView.Visible = False
            SOL_HELICEDataGridView.Visible = False
            LABEL_02.Visible = True
        Else
            ENT_HELICEDataGridView.Visible = True
            SOL_HELICEDataGridView.Visible = True
            LABEL_02.Visible = False
        End If
        If EST_QUADRADADataGridView.RowCount = 0 Then
            ENT_QUADRADADataGridView.Visible = False
            SOL_QUADRADADataGridView.Visible = False
            LABEL_03.Visible = True
        Else
            ENT_QUADRADADataGridView.Visible = True
            SOL_QUADRADADataGridView.Visible = True
            LABEL_03.Visible = False
```

```
        End If
        If EST_VAZADADataGridView.RowCount = 0 Then
            ENT_VAZADADataGridView.Visible = False
            SOL_VAZADADataGridView.Visible = False
            LABEL_04.Visible = True
        Else
            ENT_VAZADADataGridView.Visible = True
            SOL_VAZADADataGridView.Visible = True
            LABEL_04.Visible = False
        End If

        If EST_RAIZDataGridView.RowCount = 0 Then
            ENT_RAIZDataGridView.Visible = False
            SOL_RAIZDataGridView.Visible = False
            LABEL_05.Visible = True
        Else
            ENT_RAIZDataGridView.Visible = True
            SOL_RAIZDataGridView.Visible = True
            LABEL_05.Visible = False
        End If
        If EST_LAMADataGridView.RowCount = 0 Then
            ENT_LAMADataGridView.Visible = False
            SOL_LAMADataGridView.Visible = False
            LABEL_06.Visible = True
        Else
            ENT_LAMADataGridView.Visible = True
            SOL_LAMADataGridView.Visible = True
            LABEL_06.Visible = False
        End If
        If EST_MICRODataGridView.RowCount = 0 Then
            ENT_MICRODataGridView.Visible = False
            SOL_MICRODataGridView.Visible = False
            LABEL_07.Visible = True
        Else
            ENT_MICRODataGridView.Visible = True
            SOL_MICRODataGridView.Visible = True
            LABEL_07.Visible = False
        End If
    End Sub

    Private Sub RECUPERAR_DADOS_PADRAO(sender As Object, e As EventArgs) Handles
Btn_Recuperar_Dados_Padrao.Click
        Dim table As DataTable = BANCO_DE_DADOS.Tables(45)
        table.Clear()
        table.Rows.Add("ARGILA", "0,85", "0,8", "12", "0,3", "1", "12", "1", "1",
"12", "1", "1", "12", "0,85", "1,5", "12", "0,85", "0,9", "12", "1", "3", "12")
        table.Rows.Add("SILTE ARGILOSO", "0,6", "0,65", "20", "0,3", "1", "20", "1",
"1", "20", "1", "1", "20", "0,6", "1,5", "20", "0,6", "0,75", "20", "1", "3", "20")
        table.Rows.Add("SILTE ARENOSO", "0,6", "0,65", "25", "0,3", "1", "25", "1",
"1", "25", "1", "1", "25", "0,6", "1,5", "25", "0,6", "0,75", "25", "1", "3", "25")
        table.Rows.Add("AREIA", "0,5", "0,5", "40", "0,3", "1", "40", "1", "1", "40",
"1", "1", "40", "0,5", "1,5", "40", "0,5", "0,6", "40", "1", "3", "40")
        Dim table1 As DataTable = BANCO_DE_DADOS.Tables(15)
        table1.Clear()
        table1.Rows.Add("ESTACA BROCA", "1,9", "15", "1,15", "0,5")
        table1.Rows.Add("ESTACA HÉLICE CONTÍNUA", "1,8", "20", "1,15", "0,5")
        table1.Rows.Add("ESTACA PRÉ MOLDADA QUADRADA", "1,4", "40", "1,15", "0,5")
        table1.Rows.Add("ESTACA PRÉ MOLDADA VAZADA", "1,4", "40", "1,15", "0,5")
        table1.Rows.Add("ESTACA RAIZ", "1,6", "20", "1,15", "0,5")
        table1.Rows.Add("ESTACA ESCAVADA COM LAMA", "1,8", "20", "1,15", "0,5")
        table1.Rows.Add("MICROESTACA", "1,8", "20", "1,15", "0,5")
        Dim table2 As DataTable = BANCO_DE_DADOS.Tables(16)
```

```
        table2.Clear()
        table2.Rows.Add("", "ESTACA BROCA", "15", "342,11")
        table2.Rows.Add("", "ESTACA HÉLICE CONTÍNUA", "30", "342,11")
        table2.Rows.Add("", "ESTACA PRÉ MOLDADA QUADRADA", "50", "436,08")
        table2.Rows.Add("", "ESTACA PRÉ MOLDADA VAZADA", "50", "436,08")
        table2.Rows.Add("", "ESTACA RAIZ", "50", "342,11")
        table2.Rows.Add("", "ESTACA ESCAVADA COM LAMA", "50", "342,11")
        table2.Rows.Add("", "MICROESTACA", "50", "342,11")
    End Sub


    ' FUNÇÕES DE CÁLCULO
    Function CALCULO_AREA(ByRef DGV As Object, Estaca As String) As Object
        Dim diam_int, diam_ext, area, perimetro, lado As Double
        Dim i As Short
        If Estaca = "QUADRADA" Then
            For i = 0 To DGV.Rows.Count - 1
                lado = DGV.Rows(i).Cells(1).Value
                area = lado / 100 * lado / 100
                perimetro = 4 * lado / 100

                DGV.Rows(i).Cells(3).Value = Math.Round(area, 2)
                DGV.Rows(i).Cells(4).Value = Math.Round(perimetro, 2)
            Next
        ElseIf Estaca = "VAZADA" Then
            For i = 0 To DGV.Rows.Count - 1
                diam_int = DGV.Rows(i).Cells(1).Value
                diam_ext = DGV.Rows(i).Cells(2).Value
                area = ((diam_ext / 100) ^ 2 * Math.PI / 4) - ((diam_int / 100) ^ 2 *
Math.PI / 4)
                perimetro = (diam_ext / 100) * Math.PI
                DGV.Rows(i).Cells(4).Value = Math.Round(area, 2)
                DGV.Rows(i).Cells(5).Value = Math.Round(perimetro, 2)
            Next
        Else
            For i = 0 To DGV.Rows.Count - 1
                diam_ext = DGV.Rows(i).Cells(1).Value
                area = (diam_ext / 100) ^ 2 * Math.PI / 4
                perimetro = (diam_ext / 100) * Math.PI

                DGV.Rows(i).Cells(3).Value = Math.Round(area, 2)
                DGV.Rows(i).Cells(4).Value = Math.Round(perimetro, 2)
            Next
        End If
        Return DGV
    End Function
    Private Sub Decourt_Quaresma(ByVal TIPO_ESTACA As String,
                                 ByVal NFURO As Integer,
                                 ByVal max_linhas As Integer,
                                 ByVal coef_alfa As Integer,
                                 ByVal coef_beta As Integer,
                                 ByVal coef_k As Integer,
                                 ByVal ck_lateral As Object,
                                 ByVal ck_ponta As Object,
                                 ByVal ncount As Object)

        Dim area, perimetro, Media(50), Nspt2(50), N2(50), Nsoma(50), sp(50), ql(50),
resultado(50), Media_atrito(50), Pbarra1(50), Pbarra2(50), alfa, beta, coefk As Double
        Dim i, k, j, Nspt(50) As Integer
        Dim DGV, DGV_EST, DGV_COMP, DGV_TRA, DGV_COEF As DataTable
        Dim FURO_(5) As Object
        Dim narea, nperimetro As Integer
```

```
If TIPO_ESTACA = "VAZADA" Then
    narea = 4
    nperimetro = 5
Else
    narea = 3
    nperimetro = 4
End If

FURO_ = {BANCO_DE_DADOS.Tables(1),
         BANCO_DE_DADOS.Tables(2),
         BANCO_DE_DADOS.Tables(3),
         BANCO_DE_DADOS.Tables(4),
         BANCO_DE_DADOS.Tables(5),
         BANCO_DE_DADOS.Tables(6)}

DGV = FURO_(NFURO)
DGV_EST = BANCO_DE_DADOS.Tables(ET(TIPO_ESTACA, "EST"))
DGV_COMP = BANCO_DE_DADOS.Tables(ET(TIPO_ESTACA, "COMP"))
DGV_TRA = BANCO_DE_DADOS.Tables(ET(TIPO_ESTACA, "TRAC"))
DGV_COEF = BANCO_DE_DADOS.COEFICIENTES

If DGV_COMP.Rows.Count < max_linhas Then
    For i = 1 To max_linhas ' trocado 1 por 0
        DGV_COMP.Rows.Add()
        DGV_TRA.Rows.Add()
    Next
End If


For i = 0 To DGV.Rows.Count - 1
    Nspt(i) = DGV.Rows(i).Item(1)
Next i
Media(0) = (Nspt(0) + Nspt(1)) / 2

For i = 1 To DGV.Rows.Count - 1
    Media(i) = (Nspt(i - 1) + Nspt(i) + Nspt(i + 1)) / 3
Next i

For i = 0 To DGV.Rows.Count - 1
    Nspt2(i) = DGV.Rows(i).Item(1)
Next i

Nsoma(0) = Nspt2(0)
If Nsoma(0) < 3 Then Nsoma(0) = 3
If Nsoma(0) > 50 Then Nsoma(0) = 50

'calculo de N2(media ao longo do fuste)
For i = 1 To DGV.Rows.Count - 1
    If Nspt2(i) < 3 Then
        Nsoma(i) = Nsoma(i - 1) + 3
    ElseIf Nspt2(i) > 50 Then
        Nsoma(i) = Nsoma(i - 1) + 50
    Else
        Nsoma(i) = Nsoma(i - 1) + Nspt2(i)
    End If
Next i

N2(0) = 0
N2(1) = 0
For i = 2 To DGV.Rows.Count - 1
    N2(i) = Nsoma(i - 2) / (i - 1)
```

```
    Next

    'calculo de Rl
    For i = 0 To DGV.Rows.Count - 1
        Media_atrito(i) = (N2(i) / 3) + 1
    Next

    For k = 0 To DGV_EST.Rows.Count - 1
        area = DGV_EST.Rows(k).Item(narea)
        perimetro = DGV_EST.Rows(k).Item(nperimetro)

        For i = 0 To DGV.Rows.Count - 1
            For j = 0 To DGV_COEF.Rows.Count - 1
                If DGV.Rows(i).Item(2) = DGV_COEF.Rows(j).Item(0) Then
                    alfa = DGV_COEF.Rows(j).Item(coef_alfa)
                    beta = DGV_COEF.Rows(j).Item(coef_beta)
                    coefk = DGV_COEF.Rows(j).Item(coef_k)
                End If
            Next

            If ck_lateral.Checked = True Then
                ql(i) = Media_atrito(i) * perimetro * beta * i
            Else
                ql(i) = 0
            End If

            If ck_ponta.Checked = True Then
                sp(i) = alfa * area * Media(i) * coefk
            Else
                sp(i) = 0
            End If

            Pbarra1(i) = ((sp(i) + ql(i)) / 2)
            Pbarra2(i) = ((sp(i) / 4) + (ql(i) / 1.3))

            If Pbarra1(i) <= Pbarra2(i) Then
                DGV_COMP.Rows(i + ncount).Item(0) = 1 + i
                DGV_COMP.Rows(i + ncount).Item(k + 2) = Math.Round(Pbarra1(i), 2)
                DGV_TRA.Rows(i + ncount).Item(k + 2) = -Math.Round(ql(i) / 2, 2)
                DGV_TRA.Rows(i + ncount).Item(0) = 1 + i
                DGV_COMP.Rows(i + ncount).Item(1) = NFURO + 1
                DGV_TRA.Rows(i + ncount).Item(1) = NFURO + 1
            Else
                DGV_COMP.Rows(i + ncount).Item(0) = 1 + i
                DGV_COMP.Rows(i + ncount).Item(k + 2) = Math.Round(Pbarra2(i), 2)
                DGV_TRA.Rows(i + ncount).Item(k + 2) = -Math.Round(ql(i) / 2, 2)
                DGV_TRA.Rows(i + ncount).Item(0) = 1 + i
                DGV_TRA.Rows(i + ncount).Item(1) = NFURO + 1
                DGV_COMP.Rows(i + ncount).Item(1) = NFURO + 1
            End If
        Next i
    Next k
End Sub
Private Function ET(Nome As String, Tipo As String)
    Dim resp(4) As Integer
    Dim ret As Integer

    Select Case Nome
        Case "BROCA"
            resp = {7, 17, 24, 31, 38}
        Case "HELICE"
            resp = {8, 18, 27, 32, 39}
```

```
            Case "VAZADA"
                resp = {13, 19, 25, 34, 0}
            Case "LAMA"
                resp = {12, 20, 29, 33, 40}
            Case "MICRO"
                resp = {11, 21, 30, 36, 41}
            Case "QUADRADA"
                resp = {10, 22, 28, 37, 42}
            Case "RAIZ"
                resp = {9, 23, 26, 35, 43}
        End Select
        Select Case Tipo
            Case "EST"
                ret = resp(0)
            Case "ENT"
                ret = resp(1)
            Case "SOL"
                ret = resp(2)
            Case "COMP"
                ret = resp(3)
            Case "TRAC"
                ret = resp(4)
        End Select
        Return ret
    End Function
    Private Sub Altera_DGVComboBox_Rotulo_Estaca(N As Integer, Tipo_Estaca As String)
        Dim alfa As DataGridViewComboBoxColumn
        Select Case Tipo_Estaca
            Case "BROCA"
                alfa = ENT_BROCADataGridView.Columns.Item(0)
            Case "HELICE"
                alfa = ENT_HELICEDataGridView.Columns.Item(0)
            Case "VAZADA"
                alfa = ENT_VAZADADataGridView.Columns.Item(0)
            Case "LAMA"
                alfa = ENT_LAMADataGridView.Columns.Item(0)
            Case "MICRO"
                alfa = ENT_MICRODataGridView.Columns.Item(0)
            Case "QUADRADA"
                alfa = ENT_QUADRADADataGridView.Columns.Item(0)
            Case "RAIZ"
                alfa = ENT_RAIZDataGridView.Columns.Item(0)
        End Select
        alfa.Items.Clear()
        For Each k In Lista_Rotulo(N)
            alfa.Items.Add(k)
        Next
        alfa.Items.Add("")
    End Sub
    Private Function Lista_Rotulo(ByVal N As Integer)
        Dim Lista_Rot = From j In BANCO_DE_DADOS.Tables(N)
                        Select j.Item(0)
        Return Lista_Rot
    End Function
    Private Sub CALCULAR(Tipo As String)

        DEFINICOES_BLOCO(44, PILARES_CARGAS_DGV_BROCA, DEF_BLOCODataGridView,
BLOCO_2DataGridView,
                         BLOCO_3_T_1DataGridView, BLOCO_3_T_2DataGridView,
BLOCO_4DataGridView,
                         BLOCO_5_T_1DataGridView, BLOCO_5_T_2DataGridView,
BLOCO_6DataGridView,
```

```
                        BLOCO_7DataGridView, BLOCO_8DataGridView,
BLOCO_DUPLADataGridView, BLOCO_9DataGridView)

        Altera_DGVComboBox_Rotulo_Estaca(ET(Tipo, "EST"), Tipo)

        Dim a, b, c, d, f, g As Integer

        a = FURO_01DataGridView.Rows.Count
        b = a + FURO_02DataGridView.Rows.Count
        c = b + FURO_03DataGridView.Rows.Count
        d = c + FURO_04DataGridView.Rows.Count
        f = d + FURO_05DataGridView.Rows.Count
        g = f + FURO_06DataGridView.Rows.Count

        If BANCO_DE_DADOS.PILARES_CARGAS.Rows.Count > BANCO_DE_DADOS.Tables(ET(Tipo,
"ENT")).Rows.Count Or
           BANCO_DE_DADOS.PILARES_CARGAS.Rows.Count > BANCO_DE_DADOS.Tables(ET(Tipo,
"SOL")).Rows.Count Then

            While BANCO_DE_DADOS.PILARES_CARGAS.Rows.Count >
BANCO_DE_DADOS.Tables(ET(Tipo, "ENT")).Rows.Count
                BANCO_DE_DADOS.Tables(ET(Tipo, "ENT")).Rows.Add()
            End While

            While BANCO_DE_DADOS.PILARES_CARGAS.Rows.Count >
BANCO_DE_DADOS.Tables(ET(Tipo, "SOL")).Rows.Count
                BANCO_DE_DADOS.Tables(ET(Tipo, "SOL")).Rows.Add()
            End While
        End If

        Dim A_, B_, K_ As Integer
        Dim RL, RP As Object

        Dim FURO_(5) As Object
        FURO_ = {FURO_01DataGridView, FURO_02DataGridView, FURO_03DataGridView,
FURO_04DataGridView, FURO_05DataGridView, FURO_06DataGridView}

        Dim ind(5) As UInteger
        ind = {0, a, b, c, d, f}

        Select Case Tipo

            Case "BROCA"
                A_ = 1
                B_ = 2
                K_ = 3
                RL = CKB_BROCA_L
                RP = CKB_BROCA_P
                CALCULO_AREA(EST_BROCADataGridView, "BROCA")

            Case "HELICE"
                A_ = 4
                B_ = 5
                K_ = 6
                RL = CKB_HELICE_L
                RP = CKB_HELICE_P
                CALCULO_AREA(EST_HELICEDataGridView, "HELICE")

            Case "QUADRADA"
                A_ = 7
                B_ = 8
                K_ = 9
```

```
                RL = CKB_QUADRADA_L
                RP = CKB_QUADRADA_P
                CALCULO_AREA(EST_QUADRADADataGridView, "QUADRADA")

            Case "VAZADA"
                A_ = 10
                B_ = 11
                K_ = 12
                RL = CKB_VAZADA_L
                RP = CKB_VAZADA_P
                CALCULO_AREA(EST_VAZADADataGridView, "VAZADA")

            Case "RAIZ"
                A_ = 13
                B_ = 14
                K_ = 15
                RL = CKB_RAIZ_L
                RP = CKB_RAIZ_P
                CALCULO_AREA(EST_RAIZDataGridView, "RAIZ")

            Case "LAMA"
                A_ = 16
                B_ = 17
                K_ = 18
                RL = CKB_LAMA_L
                RP = CKB_LAMA_P
                CALCULO_AREA(EST_LAMADataGridView, "LAMA")
            Case "MICRO"
                A_ = 19
                B_ = 20
                K_ = 21
                CALCULO_AREA(EST_MICRODataGridView, "MICRO")
                RL = CKB_MICRO_L
                RP = CKB_MICRO_P
        End Select

        For i = 0 To 5
            If FURO_(i).Rows.Count <> 0 Then
                                Decourt_Quaresma(Tipo, i, g, A_, B_, K_, RL, RP,
ind(i))
            End If
        Next
        MsgBox("CÁLCULO EFETUADO!!!")

    End Sub

    Private Sub SOMA_TOTAL(Tipo As String)
        Select Case Tipo
            Case "BROCA"
                Dim S = (From p In BANCO_DE_DADOS.SOL_BROCA Select CType(p.CUSTO,
Decimal)).Sum()
                SOMA_BROCA.Text = S
            Case "HELICE"
                Dim S = (From p In BANCO_DE_DADOS.SOL_HELICE Select CType(p.CUSTO,
Decimal)).Sum()
                SOMA_HELICE.Text = S
            Case "QUADRADA"
                Dim S = (From p In BANCO_DE_DADOS.SOL_QUADRADA Select CType(p.CUSTO,
Decimal)).Sum()
                SOMA_QUADRADA.Text = S
            Case "VAZADA"
```

```
                Dim S = (From p In BANCO_DE_DADOS.SOL_VAZADA Select CType(p.CUSTO,
Decimal)).Sum()
                SOMA_VAZADA.Text = S
            Case "RAIZ"
                Dim S = (From p In BANCO_DE_DADOS.SOL_RAIZ Select CType(p.CUSTO,
Decimal)).Sum()
                SOMA_RAIZ.Text = S
            Case "LAMA"
                Dim S = (From p In BANCO_DE_DADOS.SOL_LAMA Select CType(p.CUSTO,
Decimal)).Sum()
                SOMA_LAMA.Text = S
            Case "MICRO"
                Dim S = (From p In BANCO_DE_DADOS.SOL_MICRO Select CType(p.CUSTO,
Decimal)).Sum()
                SOMA_MICRO.Text = S
        End Select
    End Sub
    Private Sub DEFINICOES_BLOCO(ByVal t1 As Integer, ByVal DGV_PIL As Object, ByVal
DGV_BLOCO As Object, ByVal BL2 As Object, ByVal BL3_T1 As Object, ByVal BL3_T2 As
Object, ByVal BL4 As Object, ByVal BL5_T1 As Object, ByVal BL5_T2 As Object, ByVal BL6
As Object, ByVal BL7 As Object, ByVal BL8 As Object, ByVal BL_8_T2 As Object, ByVal
BL9 As Object)
        Dim a, b, c, d, f, g, h, j, k, l, m, n As Integer
        a = BANCO_DE_DADOS.BLOCO_1.Rows.Count
        b = a + BL2.rows.count
        c = b + BL3_T1.rows.count
        d = c + BL3_T2.rows.count
        f = d + BL4.rows.count
        g = f + BL5_T1.rows.count
        h = g + BL5_T2.rows.count
        j = h + BL6.rows.count
        k = j + BL7.rows.count
        l = k + BL8.rows.count
        m = l + BL_8_T2.rows.count
        n = m + BL9.rows.count
        Dim table1 As DataTable = BANCO_DE_DADOS.Tables(t1)
        table1.Rows.Clear()
        For i = 0 To n - 1
            table1.Rows.Add()
        Next
        For i = 0 To BANCO_DE_DADOS.BLOCO_1.Rows.Count - 1
            DGV_BLOCO.rows(i).cells(0).value = "BL1"
            DGV_BLOCO.rows(i).cells(1).value = BANCO_DE_DADOS.BLOCO_1.Rows(i).Item(0)
            DGV_BLOCO.rows(i).cells(2).value = 1
            DGV_BLOCO.rows(i).cells(3).value =
Math.Round(BANCO_DE_DADOS.BLOCO_1.Rows(i).Item(1) *
BANCO_DE_DADOS.BLOCO_1.Rows(i).Item(2) * BANCO_DE_DADOS.BLOCO_1.Rows(i).Item(3), 2)
            DGV_BLOCO.rows(i).cells(4).value = 99999999
        Next
        For i = 0 To BL2.rows.count - 1
            DGV_BLOCO.rows(i + a).cells(0).value = "BL2"
            DGV_BLOCO.rows(i + a).cells(1).value = BL2.rows(i).cells(0).value
            DGV_BLOCO.rows(i + a).cells(2).value = 2
            DGV_BLOCO.rows(i + a).cells(3).value =
Math.Round(BL2.rows(i).cells(2).value * BL2.rows(i).cells(3).value *
BL2.rows(i).cells(4).value, 2)
            DGV_BLOCO.rows(i + a).cells(4).value = BL2.rows(i).cells(1).value
        Next
        For i = 0 To BL3_T1.rows.count - 1
            DGV_BLOCO.rows(i + b).cells(0).value = "BL3_T1"
            DGV_BLOCO.rows(i + b).cells(1).value = BL3_T1.rows(i).cells(0).value
            DGV_BLOCO.rows(i + b).cells(2).value = 3
```

```
            Dim D1 As Double = BL3_T1.rows(i).cells(2).value
            Dim D2 As Double = BL3_T1.rows(i).cells(3).value
            Dim D3 As Double = BL3_T1.rows(i).cells(4).value
            Dim D4 As Double = BL3_T1.rows(i).cells(5).value
            DGV_BLOCO.rows(i + b).cells(3).value = Math.Round(((D1 * D2) - (0.433 * D3
* D3) - ((D2 * 0.866 * D3) * (D1 - D3) / 2)) * D4, 2)
            DGV_BLOCO.rows(i + b).cells(4).value = BL3_T1.rows(i).cells(1).value
        Next
        For i = 0 To BL3_T2.rows.count - 1
            DGV_BLOCO.rows(i + c).cells(0).value = "BL3_T2"
            DGV_BLOCO.rows(i + c).cells(1).value = BL3_T2.rows(i).cells(0).value
            DGV_BLOCO.rows(i + c).cells(2).value = 3
            DGV_BLOCO.rows(i + c).cells(3).value =
Math.Round(BL3_T2.rows(i).cells(2).value * BL3_T2.rows(i).cells(3).value *
BL3_T2.rows(i).cells(4).value, 2)
            DGV_BLOCO.rows(i + c).cells(4).value = BL3_T2.rows(i).cells(1).value
        Next
        For i = 0 To BL4.rows.count - 1
            DGV_BLOCO.rows(i + d).cells(0).value = "BL4"
            DGV_BLOCO.rows(i + d).cells(1).value = BL4.rows(i).cells(0).value
            DGV_BLOCO.rows(i + d).cells(2).value = 4
            DGV_BLOCO.rows(i + d).cells(3).value =
Math.Round(BL4.rows(i).cells(2).value * BL4.rows(i).cells(3).value *
BL4.rows(i).cells(4).value, 2)
            DGV_BLOCO.rows(i + d).cells(4).value = BL4.rows(i).cells(1).value
        Next
        For i = 0 To BL5_T1.rows.count - 1
            DGV_BLOCO.rows(i + f).cells(0).value = "BL5_T1"
            DGV_BLOCO.rows(i + f).cells(1).value = BL5_T1.rows(i).cells(0).value
            DGV_BLOCO.rows(i + f).cells(2).value = 5
            Dim DA As Double = BL5_T1.rows(i).cells(2).value
            Dim DB As Double = BL5_T1.rows(i).cells(3).value
            Dim DC As Double = BL5_T1.rows(i).cells(4).value
            Dim DD As Double = BL5_T1.rows(i).cells(5).value
            Dim DE As Double = BL5_T1.rows(i).cells(6).value
            DGV_BLOCO.rows(i + f).cells(3).value = Math.Round(((DA * DB) - (0.433 * DC
* DC) - (((DD - DA) / 2) - DB - 0.866 * DC)) * DE, 2)
            DGV_BLOCO.rows(i + f).cells(4).value = BL5_T1.rows(i).cells(1).value
        Next
        For i = 0 To BL5_T2.rows.count - 1
            DGV_BLOCO.rows(i + g).cells(0).value = "BL5_T2"
            DGV_BLOCO.rows(i + g).cells(1).value = BL5_T2.rows(i).cells(0).value
            DGV_BLOCO.rows(i + g).cells(2).value = 5
            DGV_BLOCO.rows(i + g).cells(3).value =
Math.Round(BL5_T2.rows(i).cells(2).value * BL5_T2.rows(i).cells(3).value *
BL5_T2.rows(i).cells(4).value, 2)
            DGV_BLOCO.rows(i + g).cells(4).value = BL5_T2.rows(i).cells(1).value
        Next
        For i = 0 To BL6.rows.count - 1
            DGV_BLOCO.rows(i + h).cells(0).value = "BL6"
            DGV_BLOCO.rows(i + h).cells(1).value = BL6.rows(i).cells(0).value
            DGV_BLOCO.rows(i + h).cells(2).value = 6
            DGV_BLOCO.rows(i + h).cells(3).value =
Math.Round(BL6.rows(i).cells(2).value * BL6.rows(i).cells(3).value *
BL6.rows(i).cells(4).value, 2)
            DGV_BLOCO.rows(i + h).cells(4).value = BL6.rows(i).cells(1).value
        Next
        For i = 0 To BL7.rows.count - 1
            DGV_BLOCO.rows(i + j).cells(0).value = "BL7"
            DGV_BLOCO.rows(i + j).cells(1).value = BL7.rows(i).cells(0).value
            DGV_BLOCO.rows(i + j).cells(2).value = 7
            Dim DA1 As Double = BL7.rows(i).cells(2).value
```

```
            Dim DB1 As Double = BL7.rows(i).cells(3).value
            Dim DC1 As Double = BL7.rows(i).cells(4).value
            DGV_BLOCO.rows(i + j).cells(3).value = Math.Round((DA1 * DB1) - (0.5 * DB1
/ 2 * DB1 / 2 * 2) * DC1, 2)
            DGV_BLOCO.rows(i + j).cells(4).value = BL7.rows(i).cells(1).value
        Next
        For i = 0 To BL8.rows.count - 1
            DGV_BLOCO.rows(i + k).cells(0).value = "BL8"
            DGV_BLOCO.rows(i + k).cells(1).value = BL8.rows(i).cells(0).value
            DGV_BLOCO.rows(i + k).cells(2).value = 8
            DGV_BLOCO.rows(i + k).cells(3).value =
Math.Round(BL8.rows(i).cells(2).value * BL8.rows(i).cells(3).value *
BL8.rows(i).cells(4).value, 2)
            DGV_BLOCO.rows(i + k).cells(4).value = BL8.rows(i).cells(1).value
        Next
        For i = 0 To BL_8_T2.rows.count - 1
            DGV_BLOCO.rows(i + l).cells(0).value = "BL8_T2"
            DGV_BLOCO.rows(i + l).cells(1).value = BL_8_T2.rows(i).cells(0).value
            DGV_BLOCO.rows(i + l).cells(2).value = 8
            DGV_BLOCO.rows(i + l).cells(3).value =
Math.Round(BL_8_T2.rows(i).cells(2).value * BL_8_T2.rows(i).cells(3).value *
BL_8_T2.rows(i).cells(4).value, 2)
            DGV_BLOCO.rows(i + l).cells(4).value = BL_8_T2.rows(i).cells(1).value
        Next
        For i = 0 To BL9.rows.count - 1
            DGV_BLOCO.rows(i + m).cells(0).value = "BL9"
            DGV_BLOCO.rows(i + m).cells(1).value = BL9.rows(i).cells(0).value
            DGV_BLOCO.rows(i + m).cells(2).value = 9
            DGV_BLOCO.rows(i + m).cells(3).value =
Math.Round(BL9.rows(i).cells(2).value * BL9.rows(i).cells(3).value *
BL9.rows(i).cells(4).value, 2)
            DGV_BLOCO.rows(i + m).cells(4).value = BL9.rows(i).cells(1).value
        Next
    End Sub

    Private Sub SOLUCAO_1P(ByVal TIPO_ESTACA As String, I_PILAR As Integer, DGV_SOL As
Object)

        Dim CARGA, MOM_X, MOM_Y, YC, YS, FCK, AREA_C, AREA_S, QK, QS, TAXA, VMAX,
VMIN, E As Double
        Dim PESO, VOL_BLOCO, VOL_EST, RESIS_C, RESIS_T, FSG, FSE As Double
        Dim FURO, N_LIN, N_AREA, INI_BLO, FIM_BLO, INI_ROT, FIM_ROT, N, N_EST,
N_LIM_EMBUT, QTD_EST As Integer
        Dim CONTADOR, EMBUT_C, EMBUT_T, CONT_MENOR, EMBUTIMENTO, TESTE As Integer
        Dim ROT_BLO, ROT_EST As String

        Dim RES_1P(100000, 15) As Double
        CONTADOR = 0
        TESTE = -1
        CONT_MENOR = 99999999

        CARGA = BANCO_DE_DADOS.PILARES_CARGAS.Rows(I_PILAR).Item(1)
        MOM_X = BANCO_DE_DADOS.PILARES_CARGAS.Rows(I_PILAR).Item(2)
        MOM_Y = BANCO_DE_DADOS.PILARES_CARGAS.Rows(I_PILAR).Item(3)
        FURO = BANCO_DE_DADOS.PILARES_CARGAS.Rows(I_PILAR).Item(4)

        If IsDBNull(BANCO_DE_DADOS.PILARES_CARGAS.Rows(I_PILAR).Item(1)) Then
            CARGA = 0
        End If

        If IsDBNull(BANCO_DE_DADOS.PILARES_CARGAS.Rows(I_PILAR).Item(2)) Then
            MOM_X = 0
```

```
        End If

        If IsDBNull(BANCO_DE_DADOS.PILARES_CARGAS.Rows(I_PILAR).Item(3)) Then
            MOM_Y = 0
        End If

        'Cálculo Estrutural
        N_AREA = 3
        N_LIM_EMBUT = 2
        Select Case TIPO_ESTACA
            Case "BROCA"
                N_LIN = 0
            Case "HELICE"
                N_LIN = 1
            Case "QUADRADA"
                N_LIN = 2
            Case "VAZADA"
                N_LIN = 3
                N_AREA = 4
                N_LIM_EMBUT = 3
            Case "RAIZ"
                N_LIN = 4
            Case "ESCAVADA"
                N_LIN = 5
            Case "MICRO"
                N_LIN = 6
        End Select

        YC = BANCO_DE_DADOS.COEFICIENTE_SEGURANCA.Rows(N_LIN).Item(1)
        FCK = BANCO_DE_DADOS.COEFICIENTE_SEGURANCA.Rows(N_LIN).Item(2)
        YS = BANCO_DE_DADOS.COEFICIENTE_SEGURANCA.Rows(N_LIN).Item(3)

        If IsDBNull(BANCO_DE_DADOS.Tables(ET(TIPO_ESTACA,
"ENT")).Rows(I_PILAR).Item(0)) Then
            INI_ROT = 0
            FIM_ROT = BANCO_DE_DADOS.Tables(ET(TIPO_ESTACA, "EST")).Rows.Count - 1
        Else

            For h = 0 To BANCO_DE_DADOS.Tables(ET(TIPO_ESTACA, "EST")).Rows.Count - 1
                If BANCO_DE_DADOS.Tables(ET(TIPO_ESTACA, "EST")).Rows(h).Item(0) =
BANCO_DE_DADOS.Tables(ET(TIPO_ESTACA, "ENT")).Rows(I_PILAR).Item(0) Then
                    INI_ROT = h
                    FIM_ROT = h
                End If
            Next
        End If
        N_EST = FIM_ROT - INI_ROT

        If IsDBNull(BANCO_DE_DADOS.Tables(ET(TIPO_ESTACA,
"ENT")).Rows(I_PILAR).Item(2)) Then
            INI_BLO = 0
            FIM_BLO = BANCO_DE_DADOS.DEF_BLOCO.Rows.Count - 1
        Else
            Dim cont = 0
            For h = 0 To BANCO_DE_DADOS.DEF_BLOCO.Rows.Count - 1
                If BANCO_DE_DADOS.DEF_BLOCO.Rows(h).Item(2) =
BANCO_DE_DADOS.Tables(ET(TIPO_ESTACA, "ENT")).Rows(I_PILAR).Item(2) Then
                    If cont = 0 Then
                        INI_BLO = h
                    End If
                    cont += 1
                End If
```

```
                FIM_BLO = INI_BLO + cont - 1
            Next
        End If
        Dim PRECO As Double = 9.99E+32
        Dim N_BL = FIM_BLO - INI_BLO

        TX(vbNewLine & vbTab & String.Format("{0,-5}", "----------------------------
-------------------------------------------------------------------------------
-----------"))

        TX(vbTab & String.Format("{0,-3} {1,-8} {2,-8} {3,-8} {4,-8} {5,-8} {6,-8}
{7,-8}",
                            "PILAR : ",
                            BANCO_DE_DADOS.PILARES_CARGAS.Rows(I_PILAR).Item(0),
                            "Carga : ",
                            BANCO_DE_DADOS.PILARES_CARGAS.Rows(I_PILAR).Item(1),
                            "Mx : ",
                            BANCO_DE_DADOS.PILARES_CARGAS.Rows(I_PILAR).Item(2),
                            "My : ",
                            BANCO_DE_DADOS.PILARES_CARGAS.Rows(I_PILAR).Item(3)))

        TX(vbTab & String.Format("{0,-5}", "----------------------------------------
-------------------------------------------------------------------------
"))

        TX(vbTab & String.Format(
                        "{0,-5}|{1,-3}|{2,-8}|{3,-8}|{4,-8}|{5,-8}|{6,-8}|{7,-8}|{8,-
8}|{9,-8}|{10,-8}|{11,-8}|{12,-8}|{13,-8}|{14,-8}|",
                        "PILAR", "CON", "BLOCO", "VMAX", "VMIN", "ROT_EST", "QK", "QS",
"EMBUT", "PRECO", "QTD", "RESIS_C", "RESIS_T", "FSG", "FSE")
                        )

        For i = INI_BLO To FIM_BLO
            Dim RES(N_EST, N_BL), PRECO_CALC, PRECO_CONC, PRECO_EMBUT As Double
            PRECO_CONC = BANCO_DE_DADOS.DEFINICOES.Rows(N_LIN).Item(3)

            E = BANCO_DE_DADOS.DEF_BLOCO.Rows(i).Item(4)
            PESO = BANCO_DE_DADOS.DEF_BLOCO.Rows(i).Item(3) * 2.5
            VOL_BLOCO = BANCO_DE_DADOS.DEF_BLOCO.Rows(i).Item(3)
            ROT_BLO = BANCO_DE_DADOS.DEF_BLOCO.Rows(i).Item(0)

            If MOM_X = 0 Then
                If ROT_BLO = "BL2" Then
                    QTD_EST = 2
                    VMAX = (CARGA + PESO) / 2 + MOM_Y / E
                    VMIN = (CARGA + PESO) / 2 - MOM_Y / E
                End If
                If ROT_BLO = "BL3_T2" Then
                    QTD_EST = 3
                    VMAX = (CARGA + PESO) / 3 + MOM_Y / E
                    VMIN = (CARGA + PESO) / 3 - MOM_Y / E
                End If
                If MOM_Y = 0 Then
                    If ROT_BLO = "BL1" Then
                        QTD_EST = 1
                        VMAX = CARGA + PESO
                        VMIN = 0 + PESO
                    End If
                End If
            End If
            If ROT_BLO = "BL3_T1" Then
                QTD_EST = 3
```

```
            Dim N1 As Double
            Dim N2 As Double
            Dim N3 As Double
            Dim AUX As Double
            N1 = (CARGA + PESO) / 3 - MOM_X / (0.866 * E)
            N2 = (CARGA + PESO) / 3 + MOM_X / (1.732 * E) - MOM_Y / (E)
            N2 = (CARGA + PESO) / 3 - MOM_X / (1.732 * E) + MOM_Y / (E)
            If N1 > N2 Then
                AUX = N1
            Else
                AUX = N2
            End If
            If AUX > N3 Then
                VMAX = AUX
            Else
                VMAX = N3
            End If

            If N1 < N2 Then
                AUX = N1
            Else
                AUX = N2
            End If
            If AUX < N3 Then
                VMIN = AUX
            Else
                VMIN = N3
            End If
        End If
        If ROT_BLO = "BL4" Then
            QTD_EST = 4
            VMAX = (CARGA + PESO) / 4 + (MOM_Y + MOM_X) / (2 * E)
            VMIN = (CARGA + PESO) / 4 - (MOM_Y + MOM_X) / (2 * E)
        End If

        If ROT_BLO = "BL5_T1" Then
            'Blocos de 5 Estacas tipo 1
            QTD_EST = 5
            If MOM_X / MOM_Y >= 4.813 Then
                VMAX = (CARGA + PESO) / 5 + ((2 * MOM_X + 1.923 * MOM_Y) / (5 *
E))
                VMIN = (CARGA + PESO) / 5 - ((2 * MOM_X - 1.923 * MOM_Y) / (5 *
E))
            Else
                VMAX = (CARGA + PESO) / 5 + ((2 * MOM_X + 1.923 * MOM_Y) / (5 *
E))
                VMIN = (CARGA + PESO) / 5 - ((MOM_X + 2.89 * MOM_Y) / (5 * E))
            End If
        End If

        If ROT_BLO = "BL5_T2" Then
            'Blocos de 5 Estacas tipo 2
            QTD_EST = 5
            VMAX = (CARGA + PESO) / 5 + ((MOM_X + MOM_Y) / (2.828 * E))
            VMIN = (CARGA + PESO) / 5 - ((MOM_X + MOM_Y) / (2.828 * E))
        End If

        If ROT_BLO = "BL6" Then
            'Blocos de 6 Estacas
            QTD_EST = 6
            VMAX = (CARGA + PESO) / 6 + ((1.5 * MOM_X + 2 * MOM_Y) / (6 * E))
            VMIN = (CARGA + PESO) / 6 - ((1.5 * MOM_X + 2 * MOM_Y) / (6 * E))
```

```
            End If

            If ROT_BLO = "BL7" Then
                'Bloco de 7 estacas
                QTD_EST = 7
                If MOM_X / MOM_Y >= 0.5767 Then
                    VMAX = (CARGA + PESO) / 7 + ((1.732 * MOM_X + MOM_Y) / (6 * E))
                    VMIN = (CARGA + PESO) / 7 - ((1.732 * MOM_X + MOM_Y) / (6 * E))
                Else
                    VMAX = (CARGA + PESO) / 7 + ((MOM_Y) / (3 * E))
                    VMIN = (CARGA + PESO) / 7 - ((MOM_Y) / (3 * E))
                End If
            End If

            If ROT_BLO = "BL8" Then
                'Bloco de 8 estacas
                QTD_EST = 8
                VMAX = (CARGA + PESO) / 8 + ((1.257 * MOM_X + 1.886 * MOM_Y) / (8 *
E))
                VMIN = (CARGA + PESO) / 8 - ((1.257 * MOM_X + 1.886 * MOM_Y) / (8 *
E))
            End If

            If ROT_BLO = "BL8_T2" Then
                'Bloco de 8 estacas tipo 2
                QTD_EST = 8
                VMAX = (CARGA + PESO) / 8 + ((1.2 * MOM_X + 2 * MOM_Y) / (8 * E))
                VMIN = (CARGA + PESO) / 8 - ((1.2 * MOM_X + 2 * MOM_Y) / (8 * E))
            End If

            If ROT_BLO = "BL9" Then
                'Bloco de 9 Estacas
                QTD_EST = 9
                VMAX = (CARGA + PESO) / 9 + ((MOM_X + MOM_Y) / (6 * E))
                VMIN = (CARGA + PESO) / 9 - ((MOM_X + MOM_Y) / (6 * E))
            End If

            If VMAX = 0 Then
                  Continue For
            End If

            TX(vbTab & String.Format("{0,-5}", "-------------------------------------
-------------------------------------------------------------------------------------
---"))

            For j = INI_ROT To FIM_ROT

                ROT_EST = BANCO_DE_DADOS.Tables(ET(TIPO_ESTACA,
"EST")).rows(j).item(0)

                AREA_C = BANCO_DE_DADOS.Tables(ET(TIPO_ESTACA,
"EST")).rows(j).item(N_AREA)
                TAXA = BANCO_DE_DADOS.COEFICIENTE_SEGURANCA.Rows(N_LIN).Item(4)
                AREA_S = TAXA / 100 * AREA_C
                QK = 1 / 1.4 * (0.85 * FCK * AREA_C * 100 / YC) + (50000 * AREA_S /
YS)
                QS = (1 / 1.4 * (50000 * AREA_S / YS))

                PRECO_EMBUT = BANCO_DE_DADOS.Tables(ET(TIPO_ESTACA,
"EST")).Rows(j).Item(N_LIM_EMBUT)
```

```
                Dim ALFA = 2.5 * BANCO_DE_DADOS.Tables(ET(TIPO_ESTACA,
"EST")).Rows(j).Item(1) / 100

                If ALFA <= E Then

                    If IsDBNull(BANCO_DE_DADOS.Tables(ET(TIPO_ESTACA,
"ENT")).Rows(I_PILAR).Item(1)) Then ' Caso não seja dado um embutimeno
                        N = ET(TIPO_ESTACA, "COMP")
                        Dim ULTP_C1 As Integer

                        Dim ULTP_C As Integer = (From k In BANCO_DE_DADOS.Tables(N)
                                                 Where k.Item(1) = FURO
                                                 Select k.Item(0)).Last
                        ULTP_C1 = ULTP_C
                        Dim PROF_C As Integer = (From k In BANCO_DE_DADOS.Tables(N)
                                                 Where k.Item(1) = FURO And k.Item(j +
2) > VMAX And k.Item(0) >= 3 And k.Item(0) < ULTP_C1
                                                 Select k.Item(0))(0)

                        Dim RES_C As Double = (From k In BANCO_DE_DADOS.Tables(N)
                                               Where k.Item(1) = FURO And k.Item(j +
2) > VMAX And k.Item(0) >= 3 And k.Item(0) < ULTP_C1
                                               Select k.Item(j + 2))(0)
                        EMBUT_C = PROF_C
                        RESIS_C = RES_C


                        If VMIN <= 0 Then
                            N = ET(TIPO_ESTACA, "TRAC")
                            Dim ULTP_T1 As Integer

                            Dim ULTP_T As Integer = (From k In
BANCO_DE_DADOS.Tables(N)
                                                     Where k.Item(1) = FURO
                                                     Select k.Item(0)).Last
                            ULTP_T1 = ULTP_T

                            Dim PROF_T = (From k In BANCO_DE_DADOS.Tables(N)
                                          Where k.Item(1) = FURO And k.Item(j + 2) <
VMIN And k.Item(0) >= 3 And k.Item(0) < ULTP_T1
                                          Select k.Item(0))(0)

                            Dim RES_T As Double = (From k In BANCO_DE_DADOS.Tables(N)
                                                   Where k.Item(1) = FURO And k.Item(j
+ 2) < VMIN And k.Item(0) >= 3 And k.Item(0) < ULTP_T1
                                                   Select k.Item(j + 2))(0)
                            EMBUT_T = PROF_T
                            RESIS_T = RES_T
                        Else
                            N = ET(TIPO_ESTACA, "COMP")
                            Dim ULTP_K1 As Integer

                            Dim ULTP_K As Integer = (From k In
BANCO_DE_DADOS.Tables(N)
                                                     Where k.Item(1) = FURO
                                                     Select k.Item(0)).Last
                            ULTP_K1 = ULTP_K

                            Dim PROF_T = (From k In BANCO_DE_DADOS.Tables(N)
                                          Where k.Item(1) = FURO And k.Item(j + 2) >
VMIN And k.Item(0) >= 3 And k.Item(0) < ULTP_K1
                                          Select k.Item(0))(0)
```

```
                            Dim RES_T As Double = (From k In BANCO_DE_DADOS.Tables(N)
                                                   Where k.Item(1) = FURO And k.Item(j
+ 2) > VMIN And k.Item(0) >= 3 And k.Item(0) < ULTP_K1
                                                   Select k.Item(j + 2))(0)
                            EMBUT_T = PROF_T
                            RESIS_T = RES_T
                        End If
                        EMBUTIMENTO = Math.Max(EMBUT_C, EMBUT_T)

                    Else
                        EMBUTIMENTO = BANCO_DE_DADOS.Tables(ET(TIPO_ESTACA,
"ENT")).Rows(I_PILAR).Item(1)
                        Dim ULTP_C1 As Integer

                        N = ET(TIPO_ESTACA, "COMP")
                        Dim ULTP_C As Integer = (From k In BANCO_DE_DADOS.Tables(N)
                                                 Where k.Item(1) = FURO
                                                 Select k.Item(0)).Last
                        ULTP_C1 = ULTP_C

                        N = ET(TIPO_ESTACA, "COMP")
                        Dim RES_C As Double = (From k In BANCO_DE_DADOS.Tables(N)
                                               Where k.Item(1) = FURO And k.Item(0) =
EMBUTIMENTO And k.Item(0) >= 3
                                               Select k.Item(j + 2))(0)

                        N = ET(TIPO_ESTACA, "TRAC")
                        Dim RES_T As Double = (From k In BANCO_DE_DADOS.Tables(N)
                                               Where k.Item(1) = FURO And k.Item(0) =
EMBUTIMENTO And k.Item(0) >= 3
                                               Select k.Item(j + 2))(0)
                        RESIS_C = RES_C
                        RESIS_T = RES_T
                        If EMBUTIMENTO >= ULTP_C1 Then
                            RESIS_C = 0
                            RESIS_T = 0
                        End If

                    End If
                Else
                    RESIS_C = 0
                    RESIS_T = 0

                End If
                ' Cálculo dos Fatores de Segurança
                If (RESIS_C = 0) Then RESIS_C = 0.1
                If (RESIS_T = 0) Then RESIS_T = 0.1

                FSG = Math.Max(VMAX / RESIS_C, VMIN / RESIS_T)
                FSE = Math.Max(VMAX / QK, VMIN / QS)

                If FSG > 1 Or FSE > 1 Then
                    Continue For
                End If

                VOL_EST = AREA_C * EMBUTIMENTO
                PRECO_CALC = (VOL_BLOCO + VOL_EST) * PRECO_CONC + EMBUTIMENTO *
PRECO_EMBUT

                If PRECO_CALC < PRECO And FSE <= 1 And FSG <= 1 Then
                    PRECO = PRECO_CALC
```

```
                    CONT_MENOR = CONTADOR
                End If

                RES_1P(CONTADOR, 0) = CONTADOR
                RES_1P(CONTADOR, 1) = i
                RES_1P(CONTADOR, 2) = VMAX
                RES_1P(CONTADOR, 3) = VMIN
                RES_1P(CONTADOR, 4) = j
                RES_1P(CONTADOR, 5) = QK
                RES_1P(CONTADOR, 6) = QS
                RES_1P(CONTADOR, 7) = EMBUTIMENTO
                RES_1P(CONTADOR, 8) = PRECO_CALC
                RES_1P(CONTADOR, 9) = QTD_EST
                RES_1P(CONTADOR, 10) = RESIS_C
                RES_1P(CONTADOR, 11) = RESIS_T
                RES_1P(CONTADOR, 12) = FSG
                RES_1P(CONTADOR, 13) = FSE

                TX(vbTab &
                    String.Format(
                   "{0,-5}|{1,-3}|{2,-8}|{3,-8:###.00}|{4,-8:###.00}|{5,-8}|{6,-
8:###.00}|{7,-8:###.00}|{8,-8}|{9,-8:#####.00}|{10,-8}|{11,-8:###.00}|{12,-
8:###.00}|{13,-8:###.00}|{14,-8:###.00}|",
                    BANCO_DE_DADOS.PILARES_CARGAS.Rows(I_PILAR).Item(0),
                    RES_1P(CONTADOR, 0),
                    BANCO_DE_DADOS.DEF_BLOCO.Rows(RES_1P(CONTADOR, 1)).Item(1),
                    RES_1P(CONTADOR, 2),
                    RES_1P(CONTADOR, 3),
                    BANCO_DE_DADOS.Tables(ET(TIPO_ESTACA,
"EST")).Rows(RES_1P(CONTADOR, 4)).Item(0),
                    RES_1P(CONTADOR, 5),
                    RES_1P(CONTADOR, 6),
                    RES_1P(CONTADOR, 7),
                    RES_1P(CONTADOR, 8),
                    RES_1P(CONTADOR, 9),
                    RES_1P(CONTADOR, 10),
                    RES_1P(CONTADOR, 11),
                    RES_1P(CONTADOR, 12),
                    RES_1P(CONTADOR, 13))
                )

                CONTADOR += 1
                TESTE += 1
            Next
        Next

        TX(vbNewLine & vbTab & "RESULTADO FINAL")

        If TESTE >= 0 Then
            TX(String.Format(vbTab &
                   "{0,-5}|{1,-3}|{2,-8}|{3,-8:###.00}|{4,-8:###.00}|{5,-8}|{6,-
8:###.00}|{7,-8:###.00}|{8,-8}|{9,-8:#####.00}|{10,-8}|{11,-8:###.00}|{12,-
8:###.00}|{13,-8:###.00}|{14,-8:###.00}|",
                    BANCO_DE_DADOS.PILARES_CARGAS.Rows(I_PILAR).Item(0),
                    RES_1P(CONT_MENOR, 0),
                    BANCO_DE_DADOS.DEF_BLOCO.Rows(RES_1P(CONT_MENOR, 1)).Item(1),
                    RES_1P(CONT_MENOR, 2),
                    RES_1P(CONT_MENOR, 3),
                    BANCO_DE_DADOS.Tables(ET(TIPO_ESTACA,
"EST")).Rows(RES_1P(CONT_MENOR, 4)).Item(0),
                    RES_1P(CONT_MENOR, 5),
                    RES_1P(CONT_MENOR, 6),
```

```
                    RES_1P(CONT_MENOR, 7),
                    RES_1P(CONT_MENOR, 8),
                    RES_1P(CONT_MENOR, 9),
                    RES_1P(CONT_MENOR, 10),
                    RES_1P(CONT_MENOR, 11),
                    RES_1P(CONT_MENOR, 12),
                    RES_1P(CONT_MENOR, 13))
                )

            DGV_SOL.Rows(I_PILAR).Cells(0).Value =
BANCO_DE_DADOS.Tables(ET(TIPO_ESTACA, "EST")).Rows(RES_1P(CONT_MENOR, 4)).Item(0)
            DGV_SOL.Rows(I_PILAR).Cells(1).Value = RES_1P(CONT_MENOR, 7)
            DGV_SOL.Rows(I_PILAR).Cells(2).Value = RES_1P(CONT_MENOR, 9)
            DGV_SOL.Rows(I_PILAR).Cells(6).Value = " R$ " & Math.Round(PRECO, 2)
            DGV_SOL.Rows(I_PILAR).Cells(3).Value =
BANCO_DE_DADOS.DEF_BLOCO.Rows(RES_1P(CONT_MENOR, 1)).Item(1)
            DGV_SOL.Rows(I_PILAR).Cells(4).Value = Math.Round(RES_1P(CONT_MENOR, 12),
4) * 100 & " % "
            DGV_SOL.Rows(I_PILAR).Cells(5).Value = Math.Round(RES_1P(CONT_MENOR, 13),
4) * 100 & " % "
        Else
            TX("ERRO DE CÁLCULO")
            DGV_SOL.Rows(I_PILAR).Cells(0).Value = 0
            DGV_SOL.Rows(I_PILAR).Cells(1).Value = 0
            DGV_SOL.Rows(I_PILAR).Cells(2).Value = 0
            DGV_SOL.Rows(I_PILAR).Cells(6).Value = 10000000000.0
            DGV_SOL.Rows(I_PILAR).Cells(3).Value = 0
            DGV_SOL.Rows(I_PILAR).Cells(4).Value = 0
            DGV_SOL.Rows(I_PILAR).Cells(5).Value = 0
        End If

    End Sub

    Private Sub SOLUCAO_TODOS(ByVal TIPO_ESTACA As String)
        For i = 0 To BANCO_DE_DADOS.PILARES_CARGAS.Rows.Count - 1
            SOLUCAO_1P(TIPO_ESTACA, i, Escolha_DGV(TIPO_ESTACA, "SOL"))
        Next
    End Sub

    ' FUNÇÕES RELATIVAS A GERAÇÃO DE RELATÓRIOS
    Private Function Array_Resumo(Tipo_de_Estaca As String)

        Dim BD_SOL, BD_EST As DataTable

        Select Case Tipo_de_Estaca
            Case "BROCA"
                BD_SOL = BANCO_DE_DADOS.SOL_BROCA
                BD_EST = BANCO_DE_DADOS.EST_BROCA
            Case "HELICE"
                BD_SOL = BANCO_DE_DADOS.SOL_HELICE
                BD_EST = BANCO_DE_DADOS.EST_HELICE
            Case "RAIZ"
                BD_SOL = BANCO_DE_DADOS.SOL_RAIZ
                BD_EST = BANCO_DE_DADOS.EST_RAIZ
            Case "MICRO"
                BD_SOL = BANCO_DE_DADOS.SOL_MICRO
                BD_EST = BANCO_DE_DADOS.EST_MICRO
            Case "QUADRADA"
                BD_SOL = BANCO_DE_DADOS.SOL_QUADRADA
                BD_EST = BANCO_DE_DADOS.EST_QUADRADA
            Case "VAZADA"
                BD_SOL = BANCO_DE_DADOS.SOL_VAZADA
```

```
                BD_EST = BANCO_DE_DADOS.EST_VAZADA
            Case "LAMA"
                BD_SOL = BANCO_DE_DADOS.SOL_LAMA
                BD_EST = BANCO_DE_DADOS.EST_LAMA
        End Select
        Dim embuts = From a In (From p In BD_SOL Select CType(p(1),
Integer)).Distinct()
                     Order By a Ascending

        Dim rotulos = (From p In BD_EST Select p(0)).Distinct


        Dim RESUMO(embuts.Count, rotulos.Count) As String

        RESUMO(0, 0) = "prof./est."

        For i = 1 To rotulos.Count
            RESUMO(0, i) = rotulos(i - 1)
        Next

        For i = 1 To embuts.Count
            RESUMO(i, 0) = embuts(i - 1)
        Next

        For i = 0 To embuts.Count - 1
            For j = 0 To rotulos.Count - 1
                Dim res1 = (From p In BD_SOL Where p(0) = rotulos(j) And p(1) =
embuts(i) Select CType(p(2), Integer)).Sum()
                RESUMO(i + 1, j + 1) = res1
            Next
        Next

        Return RESUMO

    End Function

    Private Sub PDF_Resumo()

        Dim P As New Paragraph
        Dim PDFfile As New Document(PageSize.A4, 40, 40, 40, 40)
        PDFfile.AddTitle("Resumo de Quantidades de Estacas")

        Try
            Me.SaveFileDialog1.Filter = "PDF (*.pdf)|*.pdf"
            Me.SaveFileDialog1.ShowDialog()
            Dim end1 = SaveFileDialog1.FileName
            Dim Write As PdfWriter = PdfWriter.GetInstance(PDFfile, New
FileStream(end1, FileMode.Create))
            PDFfile.Open()
        Catch ex As Exception
            Dim Write As PdfWriter = PdfWriter.GetInstance(PDFfile, New
FileStream(endereco, FileMode.Create))
            PDFfile.Open()
        End Try


        Dim pTitle As New Font(iTextSharp.text.Font.FontFamily.HELVETICA, 10,
iTextSharp.text.Font.BOLD, BaseColor.BLACK)
        Dim pText As New Font(iTextSharp.text.Font.FontFamily.HELVETICA, 10,
iTextSharp.text.Font.NORMAL, BaseColor.BLACK)
```

```
        P = New Paragraph(New Chunk("CALFES - Cálculo de Fundações em Estacas por
Decourt-Quaresma", pTitle))
        P.Alignment = Element.ALIGN_CENTER
        P.SpacingAfter = 25.0F
        PDFfile.Add(P)

        P = New Paragraph(New Chunk("Resumo das Quantidades de Estacas por Embutimento
e Seção Transversal", pText))
        P.Alignment = Element.ALIGN_CENTER
        P.SpacingAfter = 10.0F
        PDFfile.Add(P)

        P = New Paragraph(New Chunk("OBRA: " & CLIENTETextBox.Text, pText))
        P.Alignment = Element.ALIGN_LEFT
        P.SpacingAfter = 5.0F
        PDFfile.Add(P)

        P = New Paragraph(New Chunk("DATA: " & MaskedTextBox1.Text, pText))
        P.Alignment = Element.ALIGN_LEFT
        P.SpacingAfter = 5.0F
        PDFfile.Add(P)

        Dim lista_estacas = {"BROCA", "HELICE", "QUADRADA", "VAZADA", "RAIZ", "LAMA",
"MICRO"}

        For Each estaca In lista_estacas

            Dim table As DataTable = BANCO_DE_DADOS.Tables(ET(estaca, "SOL"))
            Dim ncol, nlin As Integer

            If Not IsDBNull(table(0)) Then

                P = New Paragraph(New Chunk("TIPO DE ESTACA: " & estaca, pText))
                P.Alignment = Element.ALIGN_LEFT
                P.SpacingAfter = 5.0F
                PDFfile.Add(P)

                Dim RES = Array_Resumo(estaca)
                nlin = CType(RES.GetLength(0), Integer)
                ncol = CType(RES.GetLength(1), Integer)

                Dim pTable As New Font(iTextSharp.text.Font.FontFamily.HELVETICA, 10,
iTextSharp.text.Font.NORMAL, BaseColor.BLACK)

                Dim PdfTable As New PdfPTable(ncol)
                PdfTable.TotalWidth = 500.0F
                PdfTable.LockedWidth = True

                Dim widths(0 To ncol - 1) As Single
                For i As Integer = 0 To ncol - 1
                    widths(i) = 1.0F
                Next

                PdfTable.SetWidths(widths)
                PdfTable.HorizontalAlignment = 0
                PdfTable.SpacingBefore = 5.0F

                Dim pdfcell As PdfPCell = New PdfPCell

                For i As Integer = 0 To nlin - 1
                    For j As Integer = 0 To ncol - 1
                        pdfcell = New PdfPCell(New Phrase(RES(i, j), pTable))
```

```
                        pdfcell.HorizontalAlignment = PdfPCell.ALIGN_CENTER
                        PdfTable.HorizontalAlignment = PdfPCell.ALIGN_CENTER
                        PdfTable.AddCell(pdfcell)
                    Next
                Next

                PDFfile.Add(PdfTable)
                P = New Paragraph(New Chunk("", pText))
                P.Alignment = Element.ALIGN_LEFT
                P.SpacingAfter = 5.0F
                P.SpacingBefore = 5.0F
                PDFfile.Add(P)
            End If

        Next

        PDFfile.Close()

    End Sub

    Private Sub ResumoToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles ResumoToolStripMenuItem.Click
        PDF_Resumo()
    End Sub

    Private Function Array_Resultado(Tipo_de_Estaca As String)

        Dim BD_SOL, BD_EST As DataTable

        Select Case Tipo_de_Estaca
            Case "BROCA"
                BD_SOL = BANCO_DE_DADOS.SOL_BROCA
                BD_EST = BANCO_DE_DADOS.EST_BROCA
            Case "HELICE"
                BD_SOL = BANCO_DE_DADOS.SOL_HELICE
                BD_EST = BANCO_DE_DADOS.EST_HELICE
            Case "RAIZ"
                BD_SOL = BANCO_DE_DADOS.SOL_RAIZ
                BD_EST = BANCO_DE_DADOS.EST_RAIZ
            Case "MICRO"
                BD_SOL = BANCO_DE_DADOS.SOL_MICRO
                BD_EST = BANCO_DE_DADOS.EST_MICRO
            Case "QUADRADA"
                BD_SOL = BANCO_DE_DADOS.SOL_QUADRADA
                BD_EST = BANCO_DE_DADOS.EST_QUADRADA
            Case "VAZADA"
                BD_SOL = BANCO_DE_DADOS.SOL_VAZADA
                BD_EST = BANCO_DE_DADOS.EST_VAZADA
            Case "LAMA"
                BD_SOL = BANCO_DE_DADOS.SOL_LAMA
                BD_EST = BANCO_DE_DADOS.EST_LAMA
        End Select


        Dim pilares = From a In (From p In BANCO_DE_DADOS.PILARES_CARGAS Select
CType(p(0), String))
        Dim cargas = From a In (From p In BANCO_DE_DADOS.PILARES_CARGAS Select
CType(p(1), Integer))

        Dim RESULTADOS(pilares.Count, 8) As String

        RESULTADOS(0, 0) = "Pilares"
```

```
        For i = 1 To pilares.Count
            RESULTADOS(i, 0) = pilares(i - 1)
        Next

        For i = 1 To cargas.Count
            RESULTADOS(i, 1) = cargas(i - 1)
        Next

        RESULTADOS(0, 1) = "CARGAS"
        RESULTADOS(0, 2) = "ESTACA"
        RESULTADOS(0, 3) = "EMBUT."
        RESULTADOS(0, 4) = "QTD."
        RESULTADOS(0, 5) = "BLOCO"
        RESULTADOS(0, 6) = "FSD/FRD"
        RESULTADOS(0, 7) = "FSD/CEST"
        RESULTADOS(0, 8) = "CUSTO"

        For i = 0 To CType(BD_SOL.Rows.Count, Integer) - 1
            For j = 0 To CType(BD_SOL.Columns.Count, Integer) - 1
                RESULTADOS(i + 1, j + 2) = BD_SOL(i)(j).ToString

            Next
        Next

        Return RESULTADOS

    End Function

    Private Sub PDF_Resultados()

        Dim P As New Paragraph
        Dim PDFfile As New Document(PageSize.A4, 40, 40, 40, 40)
        PDFfile.AddTitle("Resumo de Quantidades de Estacas")

        Try
            Me.SaveFileDialog1.Filter = "PDF (*.pdf)|*.pdf"
            Me.SaveFileDialog1.ShowDialog()
            Dim end1 = SaveFileDialog1.FileName
            Dim Write As PdfWriter = PdfWriter.GetInstance(PDFfile, New
FileStream(end1, FileMode.Create))
            PDFfile.Open()
        Catch ex As Exception
            Dim Write As PdfWriter = PdfWriter.GetInstance(PDFfile, New
FileStream(endereco, FileMode.Create))
            PDFfile.Open()
        End Try


        Dim pTitle As New Font(iTextSharp.text.Font.FontFamily.HELVETICA, 10,
iTextSharp.text.Font.BOLD, BaseColor.BLACK)
        Dim pText As New Font(iTextSharp.text.Font.FontFamily.HELVETICA, 10,
iTextSharp.text.Font.NORMAL, BaseColor.BLACK)

        P = New Paragraph(New Chunk("CALFES - Cálculo de Fundações em Estacas por
Decourt-Quaresma", pTitle))
        P.Alignment = Element.ALIGN_CENTER
        P.SpacingAfter = 25.0F
        PDFfile.Add(P)

        P = New Paragraph(New Chunk("Resultados de Dimensionamento", pText))
        P.Alignment = Element.ALIGN_CENTER
```

```
        P.SpacingAfter = 10.0F
        PDFfile.Add(P)

        P = New Paragraph(New Chunk("OBRA: " & CLIENTETextBox.Text, pText))
        P.Alignment = Element.ALIGN_LEFT
        P.SpacingAfter = 5.0F
        PDFfile.Add(P)

        P = New Paragraph(New Chunk("DATA: " & MaskedTextBox1.Text, pText))
        P.Alignment = Element.ALIGN_LEFT
        P.SpacingAfter = 5.0F
        PDFfile.Add(P)

        Dim lista_estacas = {"BROCA", "HELICE", "QUADRADA", "VAZADA", "RAIZ", "LAMA",
"MICRO"}

        For Each estaca In lista_estacas

            Dim table As DataTable = BANCO_DE_DADOS.Tables(ET(estaca, "SOL"))
            Dim ncol, nlin As Integer

            If Not IsDBNull(table(0)) Then

                P = New Paragraph(New Chunk("TIPO DE ESTACA: " & estaca, pText))
                P.Alignment = Element.ALIGN_LEFT
                P.SpacingAfter = 5.0F
                PDFfile.Add(P)

                Dim RES = Array_Resultado(estaca)
                nlin = CType(RES.GetLength(0), Integer)
                ncol = CType(RES.GetLength(1), Integer)

                Dim pTable As New Font(iTextSharp.text.Font.FontFamily.HELVETICA, 8,
iTextSharp.text.Font.NORMAL, BaseColor.BLACK)

                Dim PdfTable As New PdfPTable(ncol)
                PdfTable.TotalWidth = 500.0F
                PdfTable.LockedWidth = True

                Dim widths(0 To ncol - 1) As Single
                For i As Integer = 0 To ncol - 1
                    widths(i) = 1.0F
                Next

                PdfTable.SetWidths(widths)
                PdfTable.HorizontalAlignment = 0
                PdfTable.SpacingBefore = 5.0F


                Dim pdfcell As PdfPCell = New PdfPCell


                For i As Integer = 0 To nlin - 1
                    For j As Integer = 0 To ncol - 1

                        pdfcell = New PdfPCell(New Phrase(RES(i, j), pTable))
                        pdfcell.HorizontalAlignment = PdfPCell.ALIGN_CENTER
                        PdfTable.HorizontalAlignment = PdfPCell.ALIGN_CENTER
                        PdfTable.AddCell(pdfcell)


                    Next
```

```
                Next


                PDFfile.Add(PdfTable)
                P = New Paragraph(New Chunk("", pText))
                P.Alignment = Element.ALIGN_LEFT
                P.SpacingAfter = 5.0F
                P.SpacingBefore = 5.0F
                PDFfile.Add(P)

                Dim ptext2 As New Font(iTextSharp.text.Font.FontFamily.HELVETICA, 8,
iTextSharp.text.Font.NORMAL, BaseColor.BLACK)
                P = New Paragraph(New Phrase("CUSTO TOTAL:  " &
lista_soma(estaca).text & "      ", ptext2))
                P.Alignment = Element.ALIGN_RIGHT
                P.SpacingAfter = 0.1F
                PDFfile.Add(P)

            End If

        Next

        PDFfile.Close()

    End Sub

    Private Sub ResultadosToolStripMenuItem_Click(sender As Object, e As EventArgs)
Handles ResultadosToolStripMenuItem.Click
        PDF_Resultados()
    End Sub

    Function APAGAR_COMP_TRAC(TIPO_ESTACA As String)
        Dim N As Integer
        Dim M As Integer
        N = ET(TIPO_ESTACA, "COMP")
        M = ET(TIPO_ESTACA, "TRAC")
        BANCO_DE_DADOS.Tables(N).Clear()
        BANCO_DE_DADOS.Tables(M).Clear()
    End Function
    ' FUNÇÕES SECUNDÁRIAS DE MANIPULAÇÃO DA INTERFACE GRÁFICA
    Private Sub CALCULAR_BROCA_Click(sender As Object, e As EventArgs) Handles
CALCULAR_BROCA.Click
        APAGAR_COMP_TRAC("BROCA")
        CALCULAR("BROCA")
        SOLUCAO_TODOS("BROCA")
        SOMA_TOTAL("BROCA")
        LOG()
    End Sub

    Private Sub CALCULAR_HELICE_Click(sender As Object, e As EventArgs) Handles
CALCULAR_HELICE.Click
        APAGAR_COMP_TRAC("HELICE")
        CALCULAR("HELICE")
        SOLUCAO_TODOS("HELICE")
        SOMA_TOTAL("HELICE")
        LOG()
    End Sub

    Private Sub CALCULAR_VAZADA_Click(sender As Object, e As EventArgs) Handles
CALCULAR_VAZADA.Click
        APAGAR_COMP_TRAC("VAZADA")
        CALCULAR("VAZADA")
```

```
        SOLUCAO_TODOS("VAZADA")
        SOMA_TOTAL("VAZADA")
        LOG()
    End Sub

    Private Sub CALCULAR_QUADRADA_Click(sender As Object, e As EventArgs) Handles
CALCULAR_QUADRADA.Click
        APAGAR_COMP_TRAC("QUADRADA")
        CALCULAR("QUADRADA")
        SOLUCAO_TODOS("QUADRADA")
        SOMA_TOTAL("QUADRADA")
        LOG()
    End Sub

    Private Sub CALCULAR_MICRO_Click(sender As Object, e As EventArgs) Handles
CALCULAR_MICRO.Click
        APAGAR_COMP_TRAC("MICRO")
        CALCULAR("MICRO")
        SOLUCAO_TODOS("MICRO")
        SOMA_TOTAL("MICRO")
        LOG()
    End Sub

    Private Sub CALCULAR_RAIZ_Click(sender As Object, e As EventArgs) Handles
CALCULAR_RAIZ.Click
        APAGAR_COMP_TRAC("RAIZ")
        CALCULAR("RAIZ")
        SOLUCAO_TODOS("RAIZ")
        SOMA_TOTAL("RAIZ")
        LOG()
    End Sub

    Private Sub CALCULAR_LAMA_Click(sender As Object, e As EventArgs) Handles
CALCULAR_LAMA.Click
        APAGAR_COMP_TRAC("LAMA")
        CALCULAR("LAMA")
        SOLUCAO_TODOS("LAMA")
        SOMA_TOTAL("LAMA")
        LOG()
    End Sub


    Function Copia_linhas(ByRef DGV As Object, ByVal e As DataGridViewCellEventArgs,
ByVal sender As Object) As Object
        'Cópia linha do combobox acima
        If (e.ColumnIndex = 3) And (sender.ToString <> "") Then
            If e.RowIndex + 1 <= DGV.Rows.Count - 1 Then
                DGV.Rows(e.RowIndex + 1).Cells(3).Value =
DGV.Rows(e.RowIndex).Cells(3).Value
            End If
        End If
        If (e.ColumnIndex = 1) Then
            If e.RowIndex + 1 <= DGV.Rows.Count - 1 Then
                DGV.Rows(e.RowIndex + 1).Cells(1).Value =
DGV.Rows(e.RowIndex).Cells(1).Value
            End If
        End If
        Return DGV
    End Function

    Private Sub travarletra_KeyPress(ByVal sender As Object, ByVal e As
KeyPressEventArgs)
```

```
        'somente numeros no datagrid view
        Dim nonNumberEntered As Boolean
        nonNumberEntered = True
        If (e.KeyChar >= Convert.ToChar(48) AndAlso e.KeyChar <= Convert.ToChar(57))
OrElse e.KeyChar = Convert.ToChar(8) Then
            nonNumberEntered = False
        End If
        If nonNumberEntered = True Then
            e.Handled = True
        Else
            e.Handled = False
        End If
    End Sub

    Private Sub numero_de_linhas_furo(ByVal textbox As Object, ByVal t As Object)
        'numero de linhas apartir da textbox
        Dim table As DataTable = BANCO_DE_DADOS.Tables(t)
        Dim i As Short
        If table.Rows.Count < textbox.Text Then
            For i = table.Rows.Count + 1 To textbox.Text
                table.Rows.Add(i)
            Next
        End If
    End Sub

    Private Sub numero_de_linhas_estacas(ByVal t As Object)
        Dim table As DataTable = BANCO_DE_DADOS.Tables(t)
        Dim newrow As DataRow = table.NewRow()
        table.Rows.Add(newrow)
    End Sub

    Private Sub apaga_linhas(ByVal t As Object)
        Dim table As DataTable = BANCO_DE_DADOS.Tables(t)
        table.Rows.Clear()
    End Sub

    Private Sub numero_de_linhas_bloco(ByVal t As Object)
        'numero de linhas apartir da textbox
        Dim table As DataTable = BANCO_DE_DADOS.Tables(t)
        If table.Rows.Count < 9 Then
            table.Rows.Add()
        End If
    End Sub

    Private Sub numero_de_linhas_pilares(ByVal t As Object)
        'numero de linhas apartir da textbox
        Dim table As DataTable = BANCO_DE_DADOS.Tables(t)
        table.Rows.Add()
    End Sub

    Private Sub numero_de_linhas_pilares1(ByVal t As Object, ByVal t2 As Object)
        'numero de linhas apartir da textbox
        Dim table As DataTable = BANCO_DE_DADOS.Tables(t)
        Dim table1 As DataTable = BANCO_DE_DADOS.Tables(t2)
        If table.Rows.Count = table1.Rows.Count Then
            table.Rows.Add()
        End If
    End Sub

    Private Sub Copia_linhas_dgv1(sender As Object, e As DataGridViewCellEventArgs)
Handles FURO_01DataGridView.CellValueChanged
        Me.FURO_01DataGridView = Copia_linhas(Me.FURO_01DataGridView, e, sender)
```

```
    End Sub

    Private Sub Copia_linhas_dgv2(sender As Object, e As DataGridViewCellEventArgs)
Handles FURO_02DataGridView.CellValueChanged
        Me.FURO_02DataGridView = Copia_linhas(Me.FURO_02DataGridView, e, sender)
    End Sub

    Private Sub Copia_linhas_dgv3(sender As Object, e As DataGridViewCellEventArgs)
Handles FURO_03DataGridView.CellValueChanged
        Me.FURO_03DataGridView = Copia_linhas(Me.FURO_03DataGridView, e, sender)
    End Sub

    Private Sub Copia_linhas_dgv4(sender As Object, e As DataGridViewCellEventArgs)
Handles FURO_04DataGridView.CellValueChanged
        Me.FURO_04DataGridView = Copia_linhas(Me.FURO_04DataGridView, e, sender)
    End Sub

    Private Sub Copia_linhas_dgv5(sender As Object, e As DataGridViewCellEventArgs)
Handles FURO_05DataGridView.CellValueChanged
        Me.FURO_05DataGridView = Copia_linhas(Me.FURO_05DataGridView, e, sender)
    End Sub

    Private Sub Copia_linhas_dgv6(sender As Object, e As DataGridViewCellEventArgs)
Handles FURO_06DataGridView.CellValueChanged
        Me.FURO_06DataGridView = Copia_linhas(Me.FURO_06DataGridView, e, sender)
    End Sub

    Private Sub Copia_linhas_dgv_1(sender As Object, e As EventArgs) Handles
FURO_01DataGridView.CurrentCellDirtyStateChanged
        'copia linha do combobox acima
        If FURO_01DataGridView.IsCurrentCellDirty Then
            FURO_01DataGridView.CommitEdit(DataGridViewDataErrorContexts.Commit)
        End If
    End Sub

    Private Sub Copia_linhas_dgv_2(sender As Object, e As EventArgs) Handles
FURO_02DataGridView.CurrentCellDirtyStateChanged
        'copia linha do combobox acima
        If FURO_02DataGridView.IsCurrentCellDirty Then
            FURO_02DataGridView.CommitEdit(DataGridViewDataErrorContexts.Commit)
        End If
    End Sub

    Private Sub Copia_linhas_dgv_3(sender As Object, e As EventArgs) Handles
FURO_03DataGridView.CurrentCellDirtyStateChanged
        'copia linha do combobox acima
        If FURO_03DataGridView.IsCurrentCellDirty Then
            FURO_03DataGridView.CommitEdit(DataGridViewDataErrorContexts.Commit)
        End If
    End Sub

    Private Sub Copia_linhas_dgv_4(sender As Object, e As EventArgs) Handles
FURO_04DataGridView.CurrentCellDirtyStateChanged
        'copia linha do combobox acima
        If FURO_04DataGridView.IsCurrentCellDirty Then
            FURO_04DataGridView.CommitEdit(DataGridViewDataErrorContexts.Commit)
        End If
    End Sub

    Private Sub Copia_linhas_dgv_5(sender As Object, e As EventArgs) Handles
FURO_05DataGridView.CurrentCellDirtyStateChanged
        'copia linha do combobox acima
```

```
        If FURO_05DataGridView.IsCurrentCellDirty Then
            FURO_05DataGridView.CommitEdit(DataGridViewDataErrorContexts.Commit)
        End If
    End Sub

    Private Sub Copia_linhas_dgv_6(sender As Object, e As EventArgs) Handles
FURO_06DataGridView.CurrentCellDirtyStateChanged
        'copia linha do combobox acima
        If FURO_06DataGridView.IsCurrentCellDirty Then
            FURO_06DataGridView.CommitEdit(DataGridViewDataErrorContexts.Commit)
        End If
    End Sub

    Private Sub Dgv_Pds_EditingControlShowing1(ByVal sender As Object, ByVal e As
System.Windows.Forms.DataGridViewEditingControlShowingEventArgs) Handles
FURO_01DataGridView.EditingControlShowing
        'somente numeros no datagrid view
        If CInt((DirectCast((sender),
System.Windows.Forms.DataGridView).CurrentCell.ColumnIndex)) = 2 Then
            AddHandler e.Control.KeyPress, AddressOf Me.travarletra_KeyPress
        End If
    End Sub

    Private Sub Txt_limite_de_sondagemKeyPress1(ByVal sender As System.Object, ByVal e
As System.Windows.Forms.KeyPressEventArgs) Handles Txt_Limite_F01.KeyPress
        'somente numeros na textbox
        If Not (Char.IsDigit(e.KeyChar) OrElse Char.IsControl(e.KeyChar)) Then
            e.Handled = True
        End If
    End Sub

    Private Sub Dgv_Pds_EditingControlShowing2(ByVal sender As Object, ByVal e As
System.Windows.Forms.DataGridViewEditingControlShowingEventArgs)
        'somente numeros no datagrid view
        If CInt((DirectCast((sender),
System.Windows.Forms.DataGridView).CurrentCell.ColumnIndex)) = 2 Then
            AddHandler e.Control.KeyPress, AddressOf Me.travarletra_KeyPress
        End If
    End Sub

    Private Sub Txt_limite_de_sondagemKeyPress2(ByVal sender As System.Object, ByVal e
As System.Windows.Forms.KeyPressEventArgs)
        'somente numeros na textbox
        If Not (Char.IsDigit(e.KeyChar) OrElse Char.IsControl(e.KeyChar)) Then
            e.Handled = True
        End If
    End Sub

    Private Sub BTF1(sender As Object, e As EventArgs) Handles Btn_Furo1.Click
        numero_de_linhas_furo(Txt_Limite_F01, 1)
        ALTERA_DGVCOMBOBOX_ROTULO_FURO(FURO_01DataGridView)
    End Sub

    Private Sub BTF2(sender As Object, e As EventArgs) Handles Btn_Furo2.Click
        numero_de_linhas_furo(Txt_Limite_F02, 2)
        ALTERA_DGVCOMBOBOX_ROTULO_FURO(FURO_02DataGridView)
    End Sub

    Private Sub BTF3(sender As Object, e As EventArgs) Handles Btn_Furo3.Click
        numero_de_linhas_furo(Txt_Limite_F03, 3)
        ALTERA_DGVCOMBOBOX_ROTULO_FURO(FURO_03DataGridView)
    End Sub
```

```
    Private Sub BTF4(sender As Object, e As EventArgs) Handles Btn_Furo4.Click
        numero_de_linhas_furo(Txt_Limite_F04, 4)
        ALTERA_DGVCOMBOBOX_ROTULO_FURO(FURO_04DataGridView)
    End Sub

    Private Sub BTF5(sender As Object, e As EventArgs) Handles Btn_Furo5.Click
        numero_de_linhas_furo(Txt_Limite_F05, 5)
        ALTERA_DGVCOMBOBOX_ROTULO_FURO(FURO_05DataGridView)
    End Sub

    Private Sub BTF6(sender As Object, e As EventArgs) Handles Btn_Furo6.Click
        numero_de_linhas_furo(Txt_Limite_F06, 6)
        ALTERA_DGVCOMBOBOX_ROTULO_FURO(FURO_06DataGridView)
    End Sub


    Private Sub Btn_Nova_broca_Click(sender As Object, e As EventArgs) Handles
Btn_Nova_broca.Click
        numero_de_linhas_estacas(7)
    End Sub

    Private Sub Btn_Nova_helice_Click(sender As Object, e As EventArgs) Handles
Btn_Nova_helice.Click
        numero_de_linhas_estacas(8)
    End Sub

    Private Sub Btn_Nova_quadrada_Click(sender As Object, e As EventArgs) Handles
Btn_Nova_quadrada.Click
        numero_de_linhas_estacas(10)
    End Sub

    Private Sub Btn_nova_vazada_Click(sender As Object, e As EventArgs) Handles
Btn_nova_vazada.Click
        numero_de_linhas_estacas(13)
    End Sub

    Private Sub Btn_nova_raiz_Click(sender As Object, e As EventArgs) Handles
Btn_nova_raiz.Click
        numero_de_linhas_estacas(9)
    End Sub

    Private Sub Btn_nova_lama_Click(sender As Object, e As EventArgs) Handles
Btn_nova_lama.Click
        numero_de_linhas_estacas(12)
    End Sub

    Private Sub Btn_nova_micro_Click(sender As Object, e As EventArgs) Handles
Btn_nova_micro.Click
        numero_de_linhas_estacas(11)
    End Sub

    Private Sub PAD_ROTULO_BROCA_MouseUp(sender As Object, e As MouseEventArgs)
Handles PAD_ROTULO_BROCA.MouseUp
        Altera_ComboBox(PAD_ROTULO_BROCA, Lista_Rotulo(ET("BROCA", "EST")))
    End Sub

    Private Sub PAD_ROTULO_RAIZ_MouseUp(sender As Object, e As MouseEventArgs) Handles
PAD_ROTULO_RAIZ.MouseUp
        Altera_ComboBox(PAD_ROTULO_RAIZ, Lista_Rotulo(ET("RAIZ", "EST")))
    End Sub
```

```
    Private Sub PAD_ROTULO_LAMA_MouseUp(sender As Object, e As MouseEventArgs) Handles
PAD_ROTULO_LAMA.MouseUp
        Altera_ComboBox(PAD_ROTULO_LAMA, Lista_Rotulo(ET("LAMA", "EST")))
    End Sub

    Private Sub PAD_ROTULO_HELICE_MouseUp(sender As Object, e As MouseEventArgs)
Handles PAD_ROTULO_HELICE.MouseUp
        Altera_ComboBox(PAD_ROTULO_HELICE, Lista_Rotulo(ET("HELICE", "EST")))
    End Sub

    Private Sub PAD_ROTULO_QUADRADA_MouseUp(sender As Object, e As MouseEventArgs)
Handles PAD_ROTULO_QUADRADA.MouseUp
        Altera_ComboBox(PAD_ROTULO_QUADRADA, Lista_Rotulo(ET("QUADRADA", "EST")))
    End Sub

    Private Sub PAD_ROTULO_VAZADA_MouseUp(sender As Object, e As MouseEventArgs)
Handles PAD_ROTULO_VAZADA.MouseUp
        Altera_ComboBox(PAD_ROTULO_VAZADA, Lista_Rotulo(ET("VAZADA", "EST")))
    End Sub

    Private Sub PAD_ROTULO_MICRO_MouseUp(sender As Object, e As MouseEventArgs)
Handles PAD_ROTULO_MICRO.MouseUp
        Altera_ComboBox(PAD_ROTULO_MICRO, Lista_Rotulo(ET("MICRO", "EST")))
    End Sub


    ' FUNÇÕES PARA PADROZINAÇÃO DE RÓTULO, EMBUTIMENTO E QUNATIDADE DE ESTACAS
#########################################################
    Private Sub Padroniza_DGV(Tipo_Estaca As String, Coluna As String)

        Dim BETA(3) As Object
        Dim COL As Integer

        For i = 0 To Escolha_DGV(Tipo_Estaca, "ENT").RowCount - 1

            Select Case Tipo_Estaca
                Case "BROCA"
                    BETA = {PAD_ROTULO_BROCA, PAD_EMBUT_BROCA, PAD_QTD_BROCA}
                Case "HELICE"
                    BETA = {PAD_ROTULO_HELICE, PAD_EMBUT_HELICE, PAD_QTD_HELICE}
                Case "VAZADA"
                    BETA = {PAD_ROTULO_VAZADA, PAD_EMBUT_VAZADA, PAD_QTD_VAZADA}
                Case "LAMA"
                    BETA = {PAD_ROTULO_LAMA, PAD_EMBUT_LAMA, PAD_QTD_LAMA}
                Case "MICRO"
                    BETA = {PAD_ROTULO_MICRO, PAD_EMBUT_MICRO, PAD_QTD_MICRO}
                Case "QUADRADA"
                    BETA = {PAD_ROTULO_QUADRADA, PAD_EMBUT_QUADRADA, PAD_QTD_QUADRADA}
                Case "RAIZ"
                    BETA = {PAD_ROTULO_RAIZ, PAD_EMBUT_RAIZ, PAD_QTD_RAIZ}
            End Select


            If Coluna = "ROTULO" Or Coluna = "BLOCO" Then
                Dim ALFA As DataGridViewComboBoxCell

                Select Case Coluna
                    Case "ROTULO"
                        ALFA = Escolha_DGV(Tipo_Estaca, "ENT").Rows(i).Cells(0)
                        COL = 0
                    Case "BLOCO"
                        ALFA = Escolha_DGV(Tipo_Estaca, "ENT").Rows(i).Cells(3)
```

```
                    COL = 3
                End Select

                If BETA(COL).Text = "" Then
                    ALFA.Value = vbNullString
                Else
                    ALFA.Value = BETA(COL).Text
                End If
            End If

            If Coluna = "EMBUT" Or Coluna = "QTD" Then
                Dim ALFA As DataGridViewCell

                Select Case Coluna
                    Case "EMBUT"
                        ALFA = Escolha_DGV(Tipo_Estaca, "ENT").Rows(i).Cells(1)
                        COL = 1
                    Case "QTD"
                        ALFA = Escolha_DGV(Tipo_Estaca, "ENT").Rows(i).Cells(2)
                        COL = 2
                End Select

                If BETA(COL).Text = "" Then
                    ALFA.Value = vbNullString
                Else
                    ALFA.Value = BETA(COL).Text
                End If
            End If
        Next

    End Sub

    Private Sub PAD_ROTULO_BROCA_SelectedIndexChanged(sender As Object, e As
EventArgs) Handles PAD_ROTULO_BROCA.SelectedIndexChanged
        Padroniza_DGV("BROCA", "ROTULO")
    End Sub

    Private Sub PAD_EMBUT_BROCA_TextChanged(sender As Object, e As EventArgs) Handles
PAD_EMBUT_BROCA.TextChanged
        Padroniza_DGV("BROCA", "EMBUT")
    End Sub

    Private Sub PAD_QTD_BROCA_TextChanged(sender As Object, e As EventArgs) Handles
PAD_QTD_BROCA.TextChanged
        Padroniza_DGV("BROCA", "QTD")
    End Sub


    Private Sub PAD_ROTULO_HELICE_SelectedIndexChanged(sender As Object, e As
EventArgs) Handles PAD_ROTULO_HELICE.SelectedIndexChanged
        Padroniza_DGV("HELICE", "ROTULO")
    End Sub

    Private Sub PAD_EMBUT_HELICE_TextChanged(sender As Object, e As EventArgs) Handles
PAD_EMBUT_HELICE.TextChanged
        Padroniza_DGV("HELICE", "EMBUT")
    End Sub

    Private Sub PAD_QTD_HELICE_TextChanged(sender As Object, e As EventArgs) Handles
PAD_QTD_HELICE.TextChanged
        Padroniza_DGV("HELICE", "QTD")
    End Sub
```

```
    Private Sub PAD_ROTULO_VAZADA_SelectedIndexChanged(sender As Object, e As
EventArgs) Handles PAD_ROTULO_VAZADA.SelectedIndexChanged
        Padroniza_DGV("VAZADA", "ROTULO")
    End Sub

    Private Sub PAD_EMBUT_VAZADA_TextChanged(sender As Object, e As EventArgs) Handles
PAD_EMBUT_VAZADA.TextChanged
        Padroniza_DGV("VAZADA", "EMBUT")
    End Sub

    Private Sub PAD_QTD_VAZADA_TextChanged(sender As Object, e As EventArgs) Handles
PAD_QTD_VAZADA.TextChanged
        Padroniza_DGV("VAZADA", "QTD")
    End Sub



    Private Sub PAD_ROTULO_QUADRADA_SelectedIndexChanged(sender As Object, e As
EventArgs) Handles PAD_ROTULO_QUADRADA.SelectedIndexChanged
        Padroniza_DGV("QUADRADA", "ROTULO")
    End Sub

    Private Sub PAD_EMBUT_QUADRADA_TextChanged(sender As Object, e As EventArgs)
Handles PAD_EMBUT_QUADRADA.TextChanged
        Padroniza_DGV("QUADRADA", "EMBUT")
    End Sub

    Private Sub PAD_QTD_QUADRADA_TextChanged(sender As Object, e As EventArgs) Handles
PAD_QTD_QUADRADA.TextChanged
        Padroniza_DGV("QUADRADA", "QTD")
    End Sub



    Private Sub PAD_ROTULO_LAMA_SelectedIndexChanged(sender As Object, e As EventArgs)
Handles PAD_ROTULO_LAMA.SelectedIndexChanged
        Padroniza_DGV("LAMA", "ROTULO")
    End Sub

    Private Sub PAD_EMBUT_LAMA_TextChanged(sender As Object, e As EventArgs) Handles
PAD_EMBUT_LAMA.TextChanged
        Padroniza_DGV("LAMA", "EMBUT")
    End Sub

    Private Sub PAD_QTD_LAMA_TextChanged(sender As Object, e As EventArgs) Handles
PAD_QTD_LAMA.TextChanged
        Padroniza_DGV("LAMA", "QTD")
    End Sub



    Private Sub PAD_ROTULO_MICRO_SelectedIndexChanged(sender As Object, e As
EventArgs) Handles PAD_ROTULO_MICRO.SelectedIndexChanged
        Padroniza_DGV("MICRO", "ROTULO")
    End Sub

    Private Sub PAD_EMBUT_MICRO_TextChanged(sender As Object, e As EventArgs) Handles
PAD_EMBUT_MICRO.TextChanged
        Padroniza_DGV("MICRO", "EMBUT")
```

```
    End Sub

    Private Sub PAD_QTD_MICRO_TextChanged(sender As Object, e As EventArgs) Handles
PAD_QTD_MICRO.TextChanged
        Padroniza_DGV("MICRO", "QTD")
    End Sub


    Private Sub PAD_ROTULO_RAIZ_SelectedIndexChanged(sender As Object, e As EventArgs)
Handles PAD_ROTULO_RAIZ.SelectedIndexChanged
        Padroniza_DGV("RAIZ", "ROTULO")
    End Sub

    Private Sub PAD_EMBUT_RAIZ_TextChanged(sender As Object, e As EventArgs) Handles
PAD_EMBUT_RAIZ.TextChanged
        Padroniza_DGV("RAIZ", "EMBUT")
    End Sub

    Private Sub PAD_QTD_RAIZ_TextChanged(sender As Object, e As EventArgs) Handles
PAD_QTD_RAIZ.TextChanged
        Padroniza_DGV("RAIZ", "QTD")
    End Sub

    Private Sub TABS_MouseClick(sender As Object, e As MouseEventArgs) Handles
TABS.MouseClick
        LIBERA_CALCULO()
    End Sub

    Private Sub TABS_MouseUp(sender As Object, e As MouseEventArgs) Handles
TABS.MouseUp
        LIBERA_CALCULO()
    End Sub


    Private Sub ADICIONA_PILAR(ByVal TIPO_ESTACA As String)

        ' Verifica se o número de linhas em ENT e SOL é igual ao número de linhas em
CARGAS_PILARES

        If BANCO_DE_DADOS.PILARES_CARGAS.Rows.Count >
BANCO_DE_DADOS.Tables(ET(TIPO_ESTACA, "ENT")).Rows.Count Or
           BANCO_DE_DADOS.PILARES_CARGAS.Rows.Count >
BANCO_DE_DADOS.Tables(ET(TIPO_ESTACA, "SOL")).Rows.Count Then

            While BANCO_DE_DADOS.PILARES_CARGAS.Rows.Count >
BANCO_DE_DADOS.Tables(ET(TIPO_ESTACA, "ENT")).Rows.Count
                BANCO_DE_DADOS.Tables(ET(TIPO_ESTACA, "ENT")).Rows.Add()
            End While

            While BANCO_DE_DADOS.PILARES_CARGAS.Rows.Count >
BANCO_DE_DADOS.Tables(ET(TIPO_ESTACA, "SOL")).Rows.Count
                BANCO_DE_DADOS.Tables(ET(TIPO_ESTACA, "SOL")).Rows.Add()
            End While
        Else
            BANCO_DE_DADOS.PILARES_CARGAS.Rows.Add()
            BANCO_DE_DADOS.Tables(ET(TIPO_ESTACA, "ENT")).Rows.Add()
            BANCO_DE_DADOS.Tables(ET(TIPO_ESTACA, "SOL")).Rows.Add()
        End If

        '@@@@
```

```
    End Sub

    Private Sub ADICIONA_PILAR_BROCA_Click(sender As Object, e As EventArgs) Handles
ADICIONA_PILAR_BROCA.Click
        ADICIONA_PILAR("BROCA")
    End Sub

    Private Sub ADICIONA_PILAR_HELICE_Click(sender As Object, e As EventArgs) Handles
ADICIONA_PILAR_HELICE.Click
        ADICIONA_PILAR("HELICE")
    End Sub

    Private Sub ADICIONA_PILAR_QUADRADA_Click(sender As Object, e As EventArgs)
Handles ADICIONA_PILAR_QUADRADA.Click
        ADICIONA_PILAR("QUADRADA")
    End Sub

    Private Sub ADICIONA_PILAR_VAZADA_Click(sender As Object, e As EventArgs) Handles
ADICIONA_PILAR_VAZADA.Click
        ADICIONA_PILAR("VAZADA")
    End Sub

    Private Sub ADICIONA_PILAR_RAIZ_Click(sender As Object, e As EventArgs) Handles
ADICIONA_PILAR_RAIZ.Click
        ADICIONA_PILAR("RAIZ")
    End Sub

    Private Sub ADICIONA_PILAR_LAMA_Click(sender As Object, e As EventArgs) Handles
ADICIONA_PILAR_LAMA.Click
        ADICIONA_PILAR("LAMA")
    End Sub

    Private Sub ADICIONA_PILAR_MICRO_Click(sender As Object, e As EventArgs) Handles
ADICIONA_PILAR_MICRO.Click
        ADICIONA_PILAR("MICRO")
    End Sub


    Private Sub APAGAR_PILAR(ByVal TIPO_ESTACA As String)

        Dim row As Integer = 0

        If TIPO_ESTACA = "BROCA" Then
            row = PILARES_CARGAS_DGV_BROCA.CurrentCell.RowIndex
            PILARES_CARGAS_DGV_BROCA.Rows.RemoveAt(row)
        End If

        If TIPO_ESTACA = "HELICE" Then
            row = PILARES_CARGAS_DGV_HELICE.CurrentCell.RowIndex
            PILARES_CARGAS_DGV_HELICE.Rows.RemoveAt(row)
        End If

        If TIPO_ESTACA = "VAZADA" Then
            row = PILARES_CARGAS_DGV_VAZADA.CurrentCell.RowIndex
            PILARES_CARGAS_DGV_VAZADA.Rows.RemoveAt(row)
        End If

        If TIPO_ESTACA = "QUADRADA" Then
            row = PILARES_CARGAS_DGV_QUADRADA.CurrentCell.RowIndex
            PILARES_CARGAS_DGV_QUADRADA.Rows.RemoveAt(row)
        End If
```

```
        If TIPO_ESTACA = "MICRO" Then
            row = PILARES_CARGAS_DGV_MICRO.CurrentCell.RowIndex
            PILARES_CARGAS_DGV_MICRO.Rows.RemoveAt(row)
        End If

        If TIPO_ESTACA = "LAMA" Then
            row = PILARES_CARGAS_DGV_LAMA.CurrentCell.RowIndex
            PILARES_CARGAS_DGV_LAMA.Rows.RemoveAt(row)
        End If

        If TIPO_ESTACA = "RAIZ" Then
            row = PILARES_CARGAS_DGV_RAIZ.CurrentCell.RowIndex
            PILARES_CARGAS_DGV_RAIZ.Rows.RemoveAt(row)
        End If

        If ENT_BROCADataGridView.Rows.Count > 0 Then
            ENT_BROCADataGridView.Rows.RemoveAt(row)
            SOL_BROCADataGridView.Rows.RemoveAt(row)
        End If
        If ENT_HELICEDataGridView.Rows.Count > 0 Then
            ENT_HELICEDataGridView.Rows.RemoveAt(row)
            SOL_HELICEDataGridView.Rows.RemoveAt(row)
        End If
        If ENT_LAMADataGridView.Rows.Count > 0 Then
            ENT_LAMADataGridView.Rows.RemoveAt(row)
            SOL_LAMADataGridView.Rows.RemoveAt(row)
        End If
        If ENT_RAIZDataGridView.Rows.Count > 0 Then
            ENT_RAIZDataGridView.Rows.RemoveAt(row)
            SOL_RAIZDataGridView.Rows.RemoveAt(row)
        End If
        If ENT_MICRODataGridView.Rows.Count > 0 Then
            ENT_MICRODataGridView.Rows.RemoveAt(row)
            SOL_MICRODataGridView.Rows.RemoveAt(row)
        End If
        If ENT_QUADRADADataGridView.Rows.Count > 0 Then
            ENT_QUADRADADataGridView.Rows.RemoveAt(row)
            SOL_QUADRADADataGridView.Rows.RemoveAt(row)
        End If
        If ENT_VAZADADataGridView.Rows.Count > 0 Then
            ENT_VAZADADataGridView.Rows.RemoveAt(row)
            SOL_VAZADADataGridView.Rows.RemoveAt(row)
        End If

    End Sub

    Private Sub APAGA_PILAR_BROCA_Click(sender As Object, e As EventArgs) Handles
APAGA_PILAR_BROCA.Click
        APAGAR_PILAR("BROCA")
    End Sub

    Private Sub APAGA_PILAR_HELICE_Click(sender As Object, e As EventArgs) Handles
APAGA_PILAR_HELICE.Click
        APAGAR_PILAR("HELICE")
    End Sub

    Private Sub APAGA_PILAR_QUADRADA_Click(sender As Object, e As EventArgs) Handles
APAGA_PILAR_QUADRADA.Click
        APAGAR_PILAR("QUADRADA")
    End Sub
```

```
    Private Sub APAGA_PILAR_VAZADA_Click(sender As Object, e As EventArgs) Handles
APAGA_PILAR_VAZADA.Click
        APAGAR_PILAR("VAZADA")
    End Sub

    Private Sub APAGA_PILAR_RAIZ_Click(sender As Object, e As EventArgs) Handles
APAGA_PILAR_RAIZ.Click
        APAGAR_PILAR("RAIZ")
    End Sub

    Private Sub APAGA_PILAR_LAMA_Click(sender As Object, e As EventArgs) Handles
APAGA_PILAR_LAMA.Click
        APAGAR_PILAR("LAMA")
    End Sub

    Private Sub APAGA_PILAR_MICRO_Click(sender As Object, e As EventArgs) Handles
APAGA_PILAR_MICRO.Click
        APAGAR_PILAR("MICRO")
    End Sub


    Private Sub Apaga1_Click(sender As Object, e As EventArgs) Handles Apaga1.Click
        apaga_linhas(1)
    End Sub

    Private Sub Apaga2_Click(sender As Object, e As EventArgs) Handles Apaga2.Click
        apaga_linhas(2)
    End Sub

    Private Sub Apaga3_Click(sender As Object, e As EventArgs) Handles Apaga3.Click
        apaga_linhas(3)
    End Sub

    Private Sub Apaga4_Click(sender As Object, e As EventArgs) Handles Apaga4.Click
        apaga_linhas(4)
    End Sub

    Private Sub Apaga5_Click(sender As Object, e As EventArgs) Handles Apaga5.Click
        apaga_linhas(5)
    End Sub

    Private Sub Apaga6_Click(sender As Object, e As EventArgs) Handles Apaga6.Click
        apaga_linhas(6)
    End Sub

    Private Sub Btn_Adc_1_Click(sender As Object, e As EventArgs) Handles
Btn_Adc_1.Click
        numero_de_linhas_bloco(58)
    End Sub
    Private Sub Btn_BL2_Click(sender As Object, e As EventArgs) Handles Btn_BL2.Click
        numero_de_linhas_bloco(46)
    End Sub

    Private Sub Btn_BL3_T1_Click(sender As Object, e As EventArgs) Handles
Btn_BL3_T1.Click
        numero_de_linhas_bloco(47)
    End Sub

    Private Sub Btn_Bl3_t2_Click(sender As Object, e As EventArgs) Handles
Btn_Bl3_t2.Click
        numero_de_linhas_bloco(48)
    End Sub
```

```
    Private Sub Btn_BL4_Click(sender As Object, e As EventArgs) Handles Btn_BL4.Click
        numero_de_linhas_bloco(49)
    End Sub

    Private Sub Btn_BL5_T1_Click(sender As Object, e As EventArgs) Handles
Btn_BL5_T1.Click
        numero_de_linhas_bloco(50)
    End Sub

    Private Sub Btn_BL5_T2_Click(sender As Object, e As EventArgs) Handles
Btn_BL5_T2.Click
        numero_de_linhas_bloco(51)
    End Sub

    Private Sub Btn_BL6_Click(sender As Object, e As EventArgs) Handles Btn_BL6.Click
        numero_de_linhas_bloco(52)
    End Sub

    Private Sub Btn_BL7_Click(sender As Object, e As EventArgs) Handles Btn_BL7.Click
        numero_de_linhas_bloco(53)
    End Sub

    Private Sub Btn_BL8_T1_Click(sender As Object, e As EventArgs) Handles
Btn_BL8_T1.Click
        numero_de_linhas_bloco(54)
    End Sub

    Private Sub Btn_BL8_T2_Click(sender As Object, e As EventArgs) Handles
Btn_BL8_T2.Click
        numero_de_linhas_bloco(56)
    End Sub

    Private Sub Btn_BL9_Click(sender As Object, e As EventArgs) Handles Btn_BL9.Click
        numero_de_linhas_bloco(55)
    End Sub



    Function lista_soma(Tipo_Estaca As Object)

        Dim alfa As Object

        Select Case Tipo_Estaca
            Case "BROCA"
                alfa = SOMA_BROCA
            Case "HELICE"
                alfa = SOMA_HELICE
            Case "VAZADA"
                alfa = SOMA_VAZADA
            Case "LAMA"
                alfa = SOMA_LAMA
            Case "MICRO"
                alfa = SOMA_MICRO
            Case "QUADRADA"
                alfa = SOMA_QUADRADA
            Case "RAIZ"
                alfa = SOMA_RAIZ
        End Select
        Return alfa
    End Function
```

```
    Private Sub Btn_Apagar_1Click(sender As Object, e As EventArgs) Handles
Btn_Apagar_1.Click
        Dim row = BLOCO_1DataGridView.CurrentCell.RowIndex
        BLOCO_1DataGridView.Rows.RemoveAt(row)
    End Sub

    Private Sub Btn_Al_2Est_Click(sender As Object, e As EventArgs) Handles
Btn_Al_2Est.Click
        Dim row = BLOCO_2DataGridView.CurrentCell.RowIndex
        BLOCO_2DataGridView.Rows.RemoveAt(row)
    End Sub

    Private Sub Btn_Al_3Est_t1_Click(sender As Object, e As EventArgs) Handles
Btn_Al_3Est_t1.Click
        Dim row = BLOCO_3_T_1DataGridView.CurrentCell.RowIndex
        BLOCO_3_T_1DataGridView.Rows.RemoveAt(row)
    End Sub

    Private Sub Btn_Al_3Est_t2_Click(sender As Object, e As EventArgs) Handles
Btn_Al_3Est_t2.Click
        Dim row = BLOCO_3_T_2DataGridView.CurrentCell.RowIndex
        BLOCO_3_T_2DataGridView.Rows.RemoveAt(row)
    End Sub

    Private Sub Btn_Al_4Est_Click(sender As Object, e As EventArgs) Handles
Btn_Al_4Est.Click
        Dim row = BLOCO_4DataGridView.CurrentCell.RowIndex
        BLOCO_4DataGridView.Rows.RemoveAt(row)
    End Sub

    Private Sub Btn_Al_5Est_t1_Click(sender As Object, e As EventArgs) Handles
Btn_Al_5Est_t1.Click
        Dim row = BLOCO_5_T_1DataGridView.CurrentCell.RowIndex
        BLOCO_5_T_1DataGridView.Rows.RemoveAt(row)
    End Sub

    Private Sub Btn_Al_5est_t2_Click(sender As Object, e As EventArgs) Handles
Btn_Al_5est_t2.Click
        Dim row = BLOCO_5_T_2DataGridView.CurrentCell.RowIndex
        BLOCO_5_T_2DataGridView.Rows.RemoveAt(row)
    End Sub

    Private Sub Btn_Al_6Est_Click(sender As Object, e As EventArgs) Handles
Btn_Al_6Est.Click
        Dim row = BLOCO_6DataGridView.CurrentCell.RowIndex
        BLOCO_6DataGridView.Rows.RemoveAt(row)
    End Sub

    Private Sub Btn_Al_7Est_Click(sender As Object, e As EventArgs) Handles
Btn_Al_7Est.Click
        Dim row = BLOCO_7DataGridView.CurrentCell.RowIndex
        BLOCO_7DataGridView.Rows.RemoveAt(row)
    End Sub

    Private Sub Btn_Al_8Est_t1_Click(sender As Object, e As EventArgs) Handles
Btn_Al_8Est_t1.Click
        Dim row = BLOCO_8DataGridView.CurrentCell.RowIndex
        BLOCO_8DataGridView.Rows.RemoveAt(row)
    End Sub
```

```
    Private Sub Btn_Al_8Est_t2_Click(sender As Object, e As EventArgs) Handles
Btn_Al_8Est_t2.Click
        Dim row = BLOCO_DUPLADataGridView.CurrentCell.RowIndex
        BLOCO_DUPLADataGridView.Rows.RemoveAt(row)
    End Sub

    Private Sub Btn_Al_9Est_Click(sender As Object, e As EventArgs) Handles
Btn_Al_9Est.Click
        Dim row = BLOCO_9DataGridView.CurrentCell.RowIndex
        BLOCO_9DataGridView.Rows.RemoveAt(row)
    End Sub

    Private Sub Btn_Al_broca_Click(sender As Object, e As EventArgs) Handles
Btn_Al_broca.Click
        Dim row = EST_BROCADataGridView.CurrentCell.RowIndex
        EST_BROCADataGridView.Rows.RemoveAt(row)
    End Sub

    Private Sub Btn_Al_helice_Click(sender As Object, e As EventArgs) Handles
Btn_Al_helice.Click
        Dim row = EST_HELICEDataGridView.CurrentCell.RowIndex
        EST_HELICEDataGridView.Rows.RemoveAt(row)
    End Sub

    Private Sub Btn_Al_quad_Click(sender As Object, e As EventArgs) Handles
Btn_Al_quad.Click
        Dim row = EST_QUADRADADataGridView.CurrentCell.RowIndex
        EST_QUADRADADataGridView.Rows.RemoveAt(row)
    End Sub

    Private Sub Btn_Al_vaz_Click(sender As Object, e As EventArgs) Handles
Btn_Al_vaz.Click
        Dim row = EST_VAZADADataGridView.CurrentCell.RowIndex
        EST_VAZADADataGridView.Rows.RemoveAt(row)
    End Sub

    Private Sub Btn_Al_raiz_Click(sender As Object, e As EventArgs) Handles
Btn_Al_raiz.Click
        Dim row = EST_RAIZDataGridView.CurrentCell.RowIndex
        EST_RAIZDataGridView.Rows.RemoveAt(row)
    End Sub

    Private Sub Btn_Al_lama_Click(sender As Object, e As EventArgs) Handles
Btn_Al_lama.Click
        Dim row = EST_LAMADataGridView.CurrentCell.RowIndex
        EST_LAMADataGridView.Rows.RemoveAt(row)
    End Sub

    Private Sub Btn_Al_micro_Click(sender As Object, e As EventArgs) Handles
Btn_Al_micro.Click
        Dim row = EST_MICRODataGridView.CurrentCell.RowIndex
        EST_MICRODataGridView.Rows.RemoveAt(row)
    End Sub

    Private Sub Btn_adl_solo_Click(sender As Object, e As EventArgs) Handles
Btn_adl_solo.Click
        Dim table As DataTable = BANCO_DE_DADOS.Tables(45)
        If table.Rows.Count = table.Rows.Count Then
            table.Rows.Add()
        End If
    End Sub
```

```
    Private Sub Btn_al_solo_Click(sender As Object, e As EventArgs) Handles
Btn_al_solo.Click
        Dim row = COEFICIENTESDataGridView.CurrentCell.RowIndex
        COEFICIENTESDataGridView.Rows.RemoveAt(row)
    End Sub

    Private Sub Altera_ComboBox(CB As Object, Lista As Object)
        CB.Items.Clear()
        For Each k In Lista
            CB.Items.Add(k)
        Next
        CB.Items.Add("")
    End Sub

    'Private Sub ADICIONA_PILAR_BROCA_Click_1(sender As Object, e As EventArgs)
Handles ADICIONA_PILAR_BROCA.Click

    'End Sub

    Function Escolha_DGV(Tipo_Estaca As String, Tipo_DGV As String)

        Dim alfa As Object

        Select Case Tipo_Estaca
            Case "BROCA"
                Select Case Tipo_DGV
                    Case "EST"
                        alfa = EST_BROCADataGridView
                    Case "ENT"
                        alfa = ENT_BROCADataGridView
                    Case "SOL"
                        alfa = SOL_BROCADataGridView
                    Case "COMP"
                        alfa = COMP_BROCADataGridView
                    Case "TRAC"
                        alfa = TRAC_BROCADataGridView
                End Select

            Case "HELICE"
                Select Case Tipo_DGV
                    Case "EST"
                        alfa = EST_HELICEDataGridView
                    Case "ENT"
                        alfa = ENT_HELICEDataGridView
                    Case "SOL"
                        alfa = SOL_HELICEDataGridView
                    Case "COMP"
                        alfa = COMP_HELICEDataGridView
                    Case "TRAC"
                        alfa = TRAC_HELICEDataGridView
                End Select

            Case "VAZADA"
                Select Case Tipo_DGV
                    Case "EST"
                        alfa = EST_VAZADADataGridView
                    Case "ENT"
                        alfa = ENT_VAZADADataGridView
                    Case "SOL"
                        alfa = SOL_VAZADADataGridView
                    Case "COMP"
                        alfa = COMP_VAZADADataGridView
```

```
                Case "TRAC"
                    alfa = TRAC_VAZADADataGridView
            End Select

        Case "LAMA"
            Select Case Tipo_DGV
                Case "EST"
                    alfa = EST_LAMADataGridView
                Case "ENT"
                    alfa = ENT_LAMADataGridView
                Case "SOL"
                    alfa = SOL_LAMADataGridView
                Case "COMP"
                    alfa = COMP_LAMADataGridView
                Case "TRAC"
                    alfa = TRAC_LAMADataGridView
            End Select

        Case "MICRO"
            Select Case Tipo_DGV
                Case "EST"
                    alfa = EST_MICRODataGridView
                Case "ENT"
                    alfa = ENT_MICRODataGridView
                Case "SOL"
                    alfa = SOL_MICRODataGridView
                Case "COMP"
                    alfa = COMP_MICRODataGridView
                Case "TRAC"
                    alfa = TRAC_MICRODataGridView
            End Select

        Case "QUADRADA"
            Select Case Tipo_DGV
                Case "EST"
                    alfa = EST_QUADRADADataGridView
                Case "ENT"
                    alfa = ENT_QUADRADADataGridView
                Case "SOL"
                    alfa = SOL_QUADRADADataGridView
                Case "COMP"
                    alfa = COMP_QUADRADADataGridView
                Case "TRAC"
                    alfa = TRAC_QUADRADADataGridView
            End Select

        Case "RAIZ"
            Select Case Tipo_DGV
                Case "EST"
                    alfa = EST_RAIZDataGridView
                Case "ENT"
                    alfa = ENT_RAIZDataGridView
                Case "SOL"
                    alfa = SOL_RAIZDataGridView
                Case "COMP"
                    alfa = COMP_RAIZDataGridView
                Case "TRAC"
                    alfa = TRAC_RAIZDataGridView
            End Select
    End Select

    Return alfa
```

```
    End Function

    Private Sub PILARES_CARGAS_DGV_BROCA_CellEndEdit(sender As Object, e As
DataGridViewCellEventArgs) Handles PILARES_CARGAS_DGV_BROCA.CellEndEdit
        Dim lin = e.RowIndex
        Dim col = e.ColumnIndex
        If (e.ColumnIndex > 0) Then
            If PILARES_CARGAS_DGV_BROCA.Rows(e.RowIndex).Cells(e.ColumnIndex).Value <
0 Then
                Dim val = CDbl(PILARES_CARGAS_DGV_BROCA.Rows(lin).Cells(col).Value)
                MessageBox.Show("Somente são permitidos valores positivos de
carregamentos!!!")
                PILARES_CARGAS_DGV_BROCA.Rows(lin).Cells(col).Value = Math.Abs(val)
            End If
        End If
    End Sub

    Private Sub PILARES_CARGAS_DGV_HELICE_CellEndEdit(sender As Object, e As
DataGridViewCellEventArgs) Handles PILARES_CARGAS_DGV_HELICE.CellEndEdit
        Dim lin = e.RowIndex
        Dim col = e.ColumnIndex
        If (e.ColumnIndex > 0) Then
            If PILARES_CARGAS_DGV_HELICE.Rows(e.RowIndex).Cells(e.ColumnIndex).Value <
0 Then
                Dim val = CDbl(PILARES_CARGAS_DGV_HELICE.Rows(lin).Cells(col).Value)
                MessageBox.Show("Somente são permitidos valores positivos de
carregamentos!!!")
                PILARES_CARGAS_DGV_HELICE.Rows(lin).Cells(col).Value = Math.Abs(val)
            End If
        End If
    End Sub

    Private Sub PILARES_CARGAS_DGV_QUADRADA_CellEndEdit(sender As Object, e As
DataGridViewCellEventArgs) Handles PILARES_CARGAS_DGV_QUADRADA.CellEndEdit
        Dim lin = e.RowIndex
        Dim col = e.ColumnIndex
        If (e.ColumnIndex > 0) Then
            If PILARES_CARGAS_DGV_QUADRADA.Rows(e.RowIndex).Cells(e.ColumnIndex).Value
< 0 Then
                Dim val = CDbl(PILARES_CARGAS_DGV_QUADRADA.Rows(lin).Cells(col).Value)
                MessageBox.Show("Somente são permitidos valores positivos de
carregamentos!!!")
                PILARES_CARGAS_DGV_QUADRADA.Rows(lin).Cells(col).Value = Math.Abs(val)
            End If
        End If
    End Sub

    Private Sub PILARES_CARGAS_DGV_VAZADA_CellEndEdit(sender As Object, e As
DataGridViewCellEventArgs) Handles PILARES_CARGAS_DGV_VAZADA.CellEndEdit
        Dim lin = e.RowIndex
        Dim col = e.ColumnIndex
        If (e.ColumnIndex > 0) Then
            If PILARES_CARGAS_DGV_VAZADA.Rows(e.RowIndex).Cells(e.ColumnIndex).Value <
0 Then
                Dim val = CDbl(PILARES_CARGAS_DGV_VAZADA.Rows(lin).Cells(col).Value)
                MessageBox.Show("Somente são permitidos valores positivos de
carregamentos!!!")
                PILARES_CARGAS_DGV_VAZADA.Rows(lin).Cells(col).Value = Math.Abs(val)
            End If
        End If
    End Sub
```

```
    Private Sub PILARES_CARGAS_DGV_RAIZ_CellEndEdit(sender As Object, e As
DataGridViewCellEventArgs) Handles PILARES_CARGAS_DGV_RAIZ.CellEndEdit
        Dim lin = e.RowIndex
        Dim col = e.ColumnIndex
        If (e.ColumnIndex > 0) Then
            If PILARES_CARGAS_DGV_RAIZ.Rows(e.RowIndex).Cells(e.ColumnIndex).Value < 0
Then
                Dim val = CDbl(PILARES_CARGAS_DGV_RAIZ.Rows(lin).Cells(col).Value)
                MessageBox.Show("Somente são permitidos valores positivos de
carregamentos!!!")
                PILARES_CARGAS_DGV_RAIZ.Rows(lin).Cells(col).Value = Math.Abs(val)
            End If
        End If
    End Sub

    Private Sub PILARES_CARGAS_DGV_LAMA_CellEndEdit(sender As Object, e As
DataGridViewCellEventArgs) Handles PILARES_CARGAS_DGV_LAMA.CellEndEdit
        Dim lin = e.RowIndex
        Dim col = e.ColumnIndex
        If (e.ColumnIndex > 0) Then
            If PILARES_CARGAS_DGV_LAMA.Rows(e.RowIndex).Cells(e.ColumnIndex).Value < 0
Then
                Dim val = CDbl(PILARES_CARGAS_DGV_LAMA.Rows(lin).Cells(col).Value)
                MessageBox.Show("Somente são permitidos valores positivos de
carregamentos!!!")
                PILARES_CARGAS_DGV_LAMA.Rows(lin).Cells(col).Value = Math.Abs(val)
            End If
        End If
    End Sub

    Private Sub PILARES_CARGAS_DGV_MICRO_CellEndEdit(sender As Object, e As
DataGridViewCellEventArgs) Handles PILARES_CARGAS_DGV_MICRO.CellEndEdit
        Dim lin = e.RowIndex
        Dim col = e.ColumnIndex
        If (e.ColumnIndex > 0) Then
            If PILARES_CARGAS_DGV_MICRO.Rows(e.RowIndex).Cells(e.ColumnIndex).Value <
0 Then
                Dim val = CDbl(PILARES_CARGAS_DGV_MICRO.Rows(lin).Cells(col).Value)
                MessageBox.Show("Somente são permitidos valores positivos de
carregamentos!!!")
                PILARES_CARGAS_DGV_MICRO.Rows(lin).Cells(col).Value = Math.Abs(val)
            End If
        End If

    End Sub

    Private Sub SobreToolStripMenuItem_Click(sender As Object, e As EventArgs) Handles
SobreToolStripMenuItem.Click
        Form2.Show()
    End Sub

End Class
```

APÊNDICE B (VALIDAÇÃO DO SOFTWARE)

Para validar a eficácia do software, foi realizado um exemplo criado para tal propósito. Para demonstrar o funcionamento do software, foram utilizados os seguintes dados:

- Perfil de sondagem (Tabela 5):

Profundidade (m)	N_{spt}	Tipo de solo
00	-	Argila
01	5	
02	6	
03	10	
04	10	
05	11	
06	12	Silte argiloso
07	14	
08	11	
09	12	
10	14	
11	21	Silte arenoso
12	24	
13	26	
14	29	
15	31	
16	35	Areia
17	37	
18	38	
19	40	
20	43	

TABELA 5 – FURO DE SONDAGEM.
FONTE: Os Autores (2016).

- Dados das propriedades geométrica das Estacas (Tabela 6):

Estaca Hélice-Contínua			
Rótulo	HE-40	HE-50	HE-60
Diâmetro (cm)	40	50	60
Preço Embutimento (R$)	59,02	72,18	90,23

TABELA 6 – DEFINIÇÕES DAS PROPRIEDADES DAS ESTACAS – HÉLICE CONTÍNUA.
FONTE: Os Autores (2016).

- Dados das propriedades dos blocos (Tabela 7, Tabela 8 e Tabela 9):

Bloco de 1 Estaca	
Rótulo	BL1
Comprimento (m)	1,00
Largura (m)	1,00
Altura (m)	1,00

TABELA 7 – DEFINIÇÕES DAS PROPRIEDADES DOS BLOCOS DE 1 ESTACA.
FONTE: Os Autores (2016).

Bloco de 2 Estacas	
Rótulo	BL2
Comprimento (m)	2,05
Largura (m)	0,80
Altura (m)	1,00
Espaçamento entre Estacas (m)	1,25

TABELA 8 – DEFINIÇÕES DAS PROPRIEDADES DOS BLOCOS DE 2 ESTACA.
FONTE: Os Autores (2016).

Bloco de 4 Estacas		
Rótulo	BL4-1	BL4-2
Comprimento (m)	1,60	3,00
Largura (m)	1,60	3,00
Altura (m)	1,10	1,50
Espaçamento entre Estacas (m)	1,00	1,50

TABELA 9 – DEFINIÇÕES DAS PROPRIEDADES DOS BLOCOS DE 4 ESTACA.
FONTE: Os Autores (2016).

- Cargas nos Pilares (Tabela 10):

Cargas			
Rótulo	P1	P2	P3
Carga Vertical (tf)	35	10	75
Momento em “X” (tfm)	0	0	15
Momento em “Y” (tfm)	12	0	17
Furo	1	1	1

TABELA 10 – DEFINIÇÕES DAS CARGAS.
FONTE: Os Autores (2016).

- Parâmetros Necessários para os cálculos (Tabela 6):
  - Coeficientes de minoração do aço e do concreto (Tabela 1);
  - Coeficientes de ponderação referentes ao método do Decourt_Quaresma (Tabelas 2 e 3);
  - Custo do metro cúbico de concreto R$342,11 para concreto de resistência de 15 a 20 Mpa e R$436,08 para concreto de resistência de 40 MPa.

A partir dos dados acima, realizou-se o primeiro cálculo. A capacidade geotécnica das estacas. Este cálculo foi realizado conforme premissas da fundamentação teórica como descrito nas tabelas abaixo.

- Cálculo da estaca com diâmetro de 40 centímetros.

Tipo de Estaca:	Hélice Contínua

ϕ =	40	cm

Prof (m)	NSPT	SOLO	k	α	β	N1	N2'	N2	Sp (m²)	Qp (tf/m²)
1	5	ARGILA	12,00	0,30	1,00	5,50	5,00	0,00	0,13	66,00
2	6	ARGILA	12,00	0,30	1,00	7,00	6,00	0,00	0,13	84,00
3	10	ARGILA	12,00	0,30	1,00	8,67	10,00	5,00	0,13	104,00
4	10	ARGILA	12,00	0,30	1,00	10,33	10,00	5,50	0,13	124,00
5	11	ARGILA	12,00	0,30	1,00	11,00	11,00	7,00	0,13	132,00
6	12	SILTE ARGILOSO	20,00	0,30	1,00	12,33	12,00	7,75	0,13	246,67
7	14	SILTE ARGILOSO	20,00	0,30	1,00	12,33	14,00	8,40	0,13	246,67
8	11	SILTE ARGILOSO	20,00	0,30	1,00	12,33	11,00	9,00	0,13	246,67
9	12	SILTE ARGILOSO	20,00	0,30	1,00	12,33	12,00	9,71	0,13	246,67
10	14	SILTE ARGILOSO	20,00	0,30	1,00	15,67	14,00	9,88	0,13	313,33
11	21	SILTE ARENOSO	25,00	0,30	1,00	19,67	21,00	10,11	0,13	491,67
12	24	SILTE ARENOSO	25,00	0,30	1,00	23,67	24,00	10,50	0,13	591,67
13	26	SILTE ARENOSO	25,00	0,30	1,00	26,33	26,00	11,45	0,13	658,33
14	29	SILTE ARENOSO	25,00	0,30	1,00	28,67	29,00	12,50	0,13	716,67
15	31	SILTE ARENOSO	25,00	0,30	1,00	31,67	31,00	13,54	0,13	791,67
16	35	AREIA	40,00	0,30	1,00	34,33	35,00	14,64	0,13	1373,33
17	37	AREIA	40,00	0,30	1,00	36,67	37,00	15,73	0,13	1466,67
18	38	AREIA	40,00	0,30	1,00	38,33	38,00	16,94	0,13	1533,33
19	40	AREIA	40,00	0,30	1,00	40,33	40,00	18,12	0,13	1613,33
20	43	AREIA	40,00	0,30	1,00	27,67	43,00	19,22	0,13	1106,67

TABELA 11 – RESULTADOS DA CAPACIDADE DE CARGA GEOTÉCNICA PARA DIÂMETRO DE 40cm – PARTE 01.
FONTE: Os Autores (2016).

Prof (m)	Rp (tf)	Sl (m)	Ql (tf/m)	Rl (tf)	Rl/2 (tf)	P/2	P	Pmax (tf)
1	2,57	1,26	1,00	0,00	0,00	1,29	0,64	0,64
2	3,28	1,26	1,00	1,26	0,63	2,27	1,79	1,79
3	4,06	1,26	2,67	6,72	3,36	5,39	6,18	5,39
4	4,84	1,26	2,83	10,71	5,36	7,77	9,45	7,77
5	5,15	1,26	3,33	16,80	8,40	10,97	14,21	10,97
6	9,62	1,26	3,58	22,58	11,29	16,10	19,77	16,10
7	9,62	1,26	3,80	28,73	14,36	19,17	24,50	19,17
8	9,62	1,26	4,00	35,28	17,64	22,45	29,54	22,45
9	9,62	1,26	4,24	42,72	21,36	26,17	35,27	26,17
10	12,22	1,26	4,29	48,67	24,33	30,44	40,49	30,44
11	19,18	1,26	4,37	55,07	27,53	37,12	47,15	37,12
12	23,08	1,26	4,50	62,37	31,19	42,72	53,75	42,72
13	25,68	1,26	4,82	72,85	36,43	49,26	62,46	49,26
14	27,95	1,26	5,17	84,63	42,32	56,29	72,09	56,29
15	30,88	1,26	5,51	97,25	48,62	64,06	82,52	64,06
16	53,56	1,26	5,88	111,15	55,58	82,36	98,89	82,36
17	57,20	1,26	6,24	125,89	62,94	91,54	111,14	91,54
18	59,80	1,26	6,65	142,35	71,18	101,08	124,45	101,08
19	62,92	1,26	7,04	159,65	79,82	111,28	138,54	111,28
20	43,16	1,26	7,41	177,33	88,67	110,25	147,20	110,25

TABELA 12 – RESULTADOS DA CAPACIDADE DE CARGA GEOTÉCNICA PARA DIÂMETRO DE 40cm – PARTE 02.
FONTE: Os Autores (2016).

- Cálculo da estaca com diâmetro de 50 centímetros.

Tipo de Estaca:	Hélice Contínua

ϕ =	50	cm

Prof (m)	NSPT	SOLO	k	α	β	N1	N2'	N2	Sp (m²)	Qp (tf/m²)
1	5	ARGILA	12,00	0,30	1,00	5,50	5,00	0,00	0,20	66,00
2	6	ARGILA	12,00	0,30	1,00	7,00	6,00	0,00	0,20	84,00
3	10	ARGILA	12,00	0,30	1,00	8,67	10,00	5,00	0,20	104,00
4	10	ARGILA	12,00	0,30	1,00	10,33	10,00	5,50	0,20	124,00
5	11	ARGILA	12,00	0,30	1,00	11,00	11,00	7,00	0,20	132,00
6	12	SILTE ARGILOSO	20,00	0,30	1,00	12,33	12,00	7,75	0,20	246,67
7	14	SILTE ARGILOSO	20,00	0,30	1,00	12,33	14,00	8,40	0,20	246,67
8	11	SILTE ARGILOSO	20,00	0,30	1,00	12,33	11,00	9,00	0,20	246,67
9	12	SILTE ARGILOSO	20,00	0,30	1,00	12,33	12,00	9,71	0,20	246,67
10	14	SILTE ARGILOSO	20,00	0,30	1,00	15,67	14,00	9,88	0,20	313,33
11	21	SILTE ARENOSO	25,00	0,30	1,00	19,67	21,00	10,11	0,20	491,67
12	24	SILTE ARENOSO	25,00	0,30	1,00	23,67	24,00	10,50	0,20	591,67
13	26	SILTE ARENOSO	25,00	0,30	1,00	26,33	26,00	11,45	0,20	658,33
14	29	SILTE ARENOSO	25,00	0,30	1,00	28,67	29,00	12,50	0,20	716,67
15	31	SILTE ARENOSO	25,00	0,30	1,00	31,67	31,00	13,54	0,20	791,67
16	35	AREIA	40,00	0,30	1,00	34,33	35,00	14,64	0,20	1373,33
17	37	AREIA	40,00	0,30	1,00	36,67	37,00	15,73	0,20	1466,67
18	38	AREIA	40,00	0,30	1,00	38,33	38,00	16,94	0,20	1533,33
19	40	AREIA	40,00	0,30	1,00	40,33	40,00	18,12	0,20	1613,33
20	43	AREIA	40,00	0,30	1,00	27,67	43,00	19,22	0,20	1106,67

TABELA 13 – RESULTADOS DA CAPACIDADE DE CARGA GEOTÉCNICA PARA DIÂMETRO DE 50cm – PARTE 01.
FONTE: Os Autores (2016).

Prof (m)	Rp (tf)	Sl (m)	Ql (tf/m)	Rl (tf)	Rl/2 (tf)	P/2	P	Pmax (tf)
1	3,96	1,57	1,00	0,00	0,00	1,98	0,99	0,99
2	5,04	1,57	1,00	1,57	0,79	3,31	2,47	2,47
3	6,24	1,57	2,67	8,37	4,19	7,31	8,00	7,31
4	7,44	1,57	2,83	13,35	6,67	10,39	12,13	10,39
5	7,92	1,57	3,33	20,93	10,47	14,43	18,08	14,43
6	14,80	1,57	3,58	28,13	14,06	21,46	25,34	21,46
7	14,80	1,57	3,80	35,80	17,90	25,30	31,24	25,30
8	14,80	1,57	4,00	43,96	21,98	29,38	37,52	29,38
9	14,80	1,57	4,24	53,23	26,62	34,02	44,65	34,02
10	18,80	1,57	4,29	60,64	30,32	39,72	51,35	39,72
11	29,50	1,57	4,37	68,61	34,31	49,06	60,16	49,06
12	35,50	1,57	4,50	77,72	38,86	56,61	68,66	56,61
13	39,50	1,57	4,82	90,77	45,39	65,14	79,70	65,14
14	43,00	1,57	5,17	105,45	52,73	74,23	91,87	74,23
15	47,50	1,57	5,51	121,17	60,59	84,34	105,08	84,34
16	82,40	1,57	5,88	138,50	69,25	110,45	127,14	110,45
17	88,00	1,57	6,24	156,86	78,43	122,43	142,66	122,43
18	92,00	1,57	6,65	177,38	88,69	134,69	159,44	134,69
19	96,80	1,57	7,04	198,93	99,46	147,86	177,22	147,86
20	66,40	1,57	7,41	220,96	110,48	143,68	186,57	143,68

TABELA 14 – RESULTADOS DA CAPACIDADE DE CARGA GEOTÉCNICA PARA DIÂMETRO DE 50cm – PARTE 02.
FONTE: Os Autores (2016).

- Cálculo da estaca com diâmetro de 60 centímetros.

Tipo de Estaca:	Hélice Contínua

ϕ =	60	cm

Prof (m)	NSPT	SOLO	C	α	β	N1	N2'	N2	Sp (m²)	Qp (tf/m²)
1	5	ARGILA	12,00	0,30	1,00	5,50	5,00	0,00	0,28	66,00
2	6	ARGILA	12,00	0,30	1,00	7,00	6,00	0,00	0,28	84,00
3	10	ARGILA	12,00	0,30	1,00	8,67	10,00	5,00	0,28	104,00
4	10	ARGILA	12,00	0,30	1,00	10,33	10,00	5,50	0,28	124,00
5	11	ARGILA	12,00	0,30	1,00	11,00	11,00	7,00	0,28	132,00
6	12	SILTE ARGILOSO	20,00	0,30	1,00	12,33	12,00	7,75	0,28	246,67
7	14	SILTE ARGILOSO	20,00	0,30	1,00	12,33	14,00	8,40	0,28	246,67
8	11	SILTE ARGILOSO	20,00	0,30	1,00	12,33	11,00	9,00	0,28	246,67
9	12	SILTE ARGILOSO	20,00	0,30	1,00	12,33	12,00	9,71	0,28	246,67
10	14	SILTE ARGILOSO	20,00	0,30	1,00	15,67	14,00	9,88	0,28	313,33
11	21	SILTE ARENOSO	25,00	0,30	1,00	19,67	21,00	10,11	0,28	491,67
12	24	SILTE ARENOSO	25,00	0,30	1,00	23,67	24,00	10,50	0,28	591,67
13	26	SILTE ARENOSO	25,00	0,30	1,00	26,33	26,00	11,45	0,28	658,33
14	29	SILTE ARENOSO	25,00	0,30	1,00	28,67	29,00	12,50	0,28	716,67
15	31	SILTE ARENOSO	25,00	0,30	1,00	31,67	31,00	13,54	0,28	791,67
16	35	AREIA	40,00	0,30	1,00	34,33	35,00	14,64	0,28	1373,33
17	37	AREIA	40,00	0,30	1,00	36,67	37,00	15,73	0,28	1466,67
18	38	AREIA	40,00	0,30	1,00	38,33	38,00	16,94	0,28	1533,33
19	40	AREIA	40,00	0,30	1,00	40,33	40,00	18,12	0,28	1613,33
20	43	AREIA	40,00	0,30	1,00	27,67	43,00	19,22	0,28	1106,67

TABELA 15 – RESULTADOS DA CAPACIDADE DE CARGA GEOTÉCNICA PARA DIÂMETRO DE 60cm – PARTE 01.
FONTE: Os Autores (2016).

Prof (m)	Rp (tf)	Sl (m)	Ql (tf/m)	Rl (tf)	Rl/2 (tf)	P/2	P	Pmax (tf)
1	5,54	1,88	1,00	0,00	0,00	2,77	1,39	1,39
2	7,06	1,88	1,00	1,88	0,94	4,47	3,21	3,21
3	8,74	1,88	2,67	10,03	5,01	9,38	9,90	9,38
4	10,42	1,88	2,83	15,98	7,99	13,20	14,90	13,20
5	11,09	1,88	3,33	25,07	12,53	18,08	22,05	18,08
6	20,72	1,88	3,58	33,68	16,84	27,20	31,09	27,20
7	20,72	1,88	3,80	42,86	21,43	31,79	38,15	31,79
8	20,72	1,88	4,00	52,64	26,32	36,68	45,67	36,68
9	20,72	1,88	4,24	63,74	31,87	42,23	54,21	42,23
10	26,32	1,88	4,29	72,62	36,31	49,47	62,44	49,47
11	41,30	1,88	4,37	82,16	41,08	61,73	73,53	61,73
12	49,70	1,88	4,50	93,06	46,53	71,38	84,01	71,38
13	55,30	1,88	4,82	108,70	54,35	82,00	97,44	82,00
14	60,20	1,88	5,17	126,27	63,14	93,24	112,18	93,24
15	66,50	1,88	5,51	145,10	72,55	105,80	128,24	105,80
16	115,36	1,88	5,88	165,84	82,92	140,60	156,41	140,60
17	123,20	1,88	6,24	187,83	93,92	155,52	175,29	155,52
18	128,80	1,88	6,65	212,40	106,20	170,60	195,59	170,60
19	135,52	1,88	7,04	238,21	119,10	186,86	217,12	186,86
20	92,96	1,88	7,41	264,59	132,30	178,78	226,77	178,78

TABELA 16 – RESULTADOS DA CAPACIDADE DE CARGA GEOTÉCNICA PARA DIÂMETRO DE 60cm – PARTE 02.
FONTE: Os Autores (2016).

Próxima etapa é cálculo das solicitações axiais máximas e mínimas para os blocos descritas nas tabelas 17, 18 e 19.

Pilar P1								
Rótulo Bloco	Distancia Entre Estacas (m)	Volume do Bloco (m^3)	Peso do Bloco (tf)	Carga Vertical (tf)	Momento na direção X (tfm)	Momento na direção Y (tfm)	Vmáx (tf)	Vmín(tf)
BL1	-	1,00	2,50	35	0	12	*	*
BL2	1,25	1,64	4,10	35	0	12	29,15	9,95
BL4-1	1,00	2,82	7,04	35	0	12	16,51	4,51
BL4-2	1,50	13,50	33,75	35	0	12	21,19	13,19
* O Software não calcula. Há um momento e não há distância entre estacas para transforma-lo em carga axial.								

TABELA 17 – SOLICITAÇÕES MÁXIMAS E MÍNIMAS – PILAR P1.
FONTE: Os Autores (2016).

Pilar P2								
Rótulo Bloco	Distancia Entre Estacas (m)	Volume do Bloco (m^3)	Peso do Bloco (tf)	Carga Vertical (tf)	Momento na direção X (tfm)	Momento na direção Y (tfm)	Vmáx (tf)	Vmín(tf)
BL1	-	1,00	2,50	10	0	0	12,50	2,50
BL2	1,25	1,64	4,10	10	0	0	7,05	7,05
BL4-1	1,00	2,82	7,04	10	0	0	4,26	4,26
BL4-2	1,50	13,50	33,75	10	0	0	10,94	10,94

TABELA 18 – SOLICITAÇÕES MÁXIMAS E MÍNIMAS – PILAR P2.
FONTE: Os Autores (2016).

Pilar P3								
Rótulo Bloco	Distancia Entre Estacas (m)	Volume do Bloco (m^3)	Peso do Bloco (tf)	Carga Vertical (tf)	Momento na direção X (tfm)	Momento na direção Y (tfm)	Vmáx (tf)	Vmín(tf)
BL1	-	1,00	2,50	75	15	17	*	*
BL2	1,25	1,64	4,10	75	15	17	**	**
BL4-1	1,00	2,82	7,04	75	15	17	36,51	4,51
BL4-2	1,50	13,50	33,75	75	15	17	37,85	16,52
* O Software não calcula. Há um momento e não há distância entre estacas para transforma-lo em carga axial.								
** O Software não calcula. Há um momento na direção x e o bloco não reconhece momento na direção x, apenas na direção Y								

TABELA 19 – SOLICITAÇÕES MÁXIMAS E MÍNIMAS – PILAR P3.
FONTE: Os Autores (2016).

Possuindo os valores de Vmáx e Vmín são comparados com a capacidade estrutural (Vide fundamentação teórica) das estacas, conforme descrito na tabela 20.

Rótulo	Capacidade Estrutural a Compressão	Capacidade Estrutural a Tração
HE-40	115,96	20,19
HE-50	178,4	31,06
HE-60	249,76	43,48

TABELA 20 – CAPACIDADE ESTRUTURAL A COMPRESSÃO E A TRAÇÃO.
FONTE: Os Autores (2016).

Para a capacidade estrutural a solicitação axial de tração foi considerando apenas a parcela resistente do aço.

Verificou-se que a capacidade estrutural das estacas é maior que a solicitação axial.

A partir desta verificação procurou-se o menor embutimento possível para a estaca para que a capacidade resistente geotécnica seja maior que a força solicitante, apresentados nas tabelas 21, 22, 23 e 24.

BLOCO1			
Pilar P1	ϕ 40 cm	ϕ 50 cm	ϕ 60 cm
Embutimento Mín	-	-	-
Pilar P2	ϕ 40 cm	ϕ 50 cm	ϕ 60 cm
Embutimento Mín	6	5	4
Pilar P3	ϕ 40 cm	ϕ 50 cm	ϕ 60 cm
Embutimento Mín	-	-	-

TABELA 21 – EMBUTIMENTOS MÍNIMOS PARA BLOCO1.
FONTE: Os Autores (2016).

BLOCO2			
Pilar P1	ϕ 40 cm	ϕ 50 cm	ϕ 60 cm
Embutimento Mín	10	8	*
Pilar P2	ϕ 40 cm	ϕ 50 cm	ϕ 60 cm
Embutimento Mín	4	3	*
Pilar P3	ϕ 40 cm	ϕ 50 cm	ϕ 60 cm
Embutimento Mín	-	-	*

TABELA 22 – EMBUTIMENTOS MÍNIMOS PARA BLOCO2.
FONTE: Os Autores (2016).

BLOCO4-1			
Pilar P1	ϕ 40 cm	ϕ 50 cm	ϕ 60 cm
Embutimento Mín	7	*	*
Pilar P2	ϕ 40 cm	ϕ 50 cm	ϕ 60 cm
Embutimento Mín	3	*	*
Pilar P3	ϕ 40 cm	ϕ 50 cm	ϕ 60 cm
Embutimento Mín	11	*	*
* Não passa na verificação, pois o espaçamento entre estacas não respeita 2,5 vezes o diâmetro da mesma.			

TABELA 23 – EMBUTIMENTOS MÍNIMOS PARA BLOCO4-1.
FONTE: Os Autores (2016).

BLOCO4-2			
Pilar P1	ϕ 40 cm	ϕ 50 cm	ϕ 60 cm
Embutimento Mín	7	6	6
Pilar P2	ϕ 40 cm	ϕ 50 cm	ϕ 60 cm
Embutimento Mín	5	5	4
Pilar P3	ϕ 40 cm	ϕ 50 cm	ϕ 60 cm
Embutimento Mín	12	10	9

TABELA 24 – EMBUTIMENTOS MÍNIMOS PARA BLOCO4-2.
FONTE: Os Autores (2016).

A partir da obtenção de todos os dados para o cálculo do custo total, foi calculado para todas as soluções possíveis um custo, e selecionado o menor custo, como demonstra a tabela 25.

	Rótulo do Bloco	Volume do Bloco (m^3)	Rótulo da Estaca	Área da seção transversal da estaca (m^2)	Embuti-mento mínimo (m)	Custo do Concreto (R$/$m^3$)	Custo do Embutimento (R$/m)	Custo Total (R$)	Menor Custo (R$)
PILAR P1	BL2	1,64	HE-40	0,13	10,00	342,11	59,02	1596,00	1596,00
	BL2	1,64	HE-50	0,20	8,00	342,11	72,18	1685,88	
	BL4-1	2,82	HE-40	0,13	7,00	342,11	59,02	1689,21	
	BL4-2	13,50	HE-40	0,13	7,00	342,11	59,02	5342,95	
	BL4-2	13,50	HE-50	0,20	6,00	342,11	72,18	5462,10	
	BL4-2	13,50	HE-60	0,28	6,00	342,11	90,23	5734,61	
PILAR P2	BL-1	1,00	HE-40	0,13	6,00	342,11	59,02	963,08	963,08
	BL-1	1,00	HE-50	0,20	5,00	342,11	72,18	1045,12	
	BL-1	1,00	HE-60	0,28	4,00	342,11	90,23	1086,19	
	BL2	1,64	HE-40	0,13	4,00	342,11	59,02	975,04	
	BL2	1,64	HE-50	0,20	3,00	342,11	72,18	982,87	
	BL4-1	2,82	HE-40	0,13	3,00	342,11	59,02	1275,23	
	BL4-2	13,50	HE-40	0,13	5,00	342,11	59,02	5135,96	
	BL4-2	13,50	HE-50	0,20	5,00	342,11	72,18	5321,50	
	BL4-2	13,50	HE-60	0,28	4,00	342,11	90,23	5362,57	
PILAR P3	BL4-1	2,82	HE-40	0,13	11,00	342,11	59,02	2103,19	2103,19
	BL4-2	13,50	HE-40	0,13	12,00	342,11	59,02	5860,42	
	BL4-2	13,50	HE-50	0,20	10,00	342,11	72,18	6024,51	
	BL4-2	13,50	HE-60	0,28	9,00	342,11	90,23	6292,67	

TABELA 25 – RESULTADOS.
FONTE: Os Autores (2016).